

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

Welcome to Rhino!

Rhino is an infrastructure for building applications that configure, manage, and monitor hardware and software. Rhino provides a common, consistent, task-based, localized, secure graphical user interface (GUI), with built-in command-line interfaces (CLIs) that system administrators can use to write scripts. Rhino applications consist of two parts:

- **Client-side GUI in Java.** The GUI runs on any platform that has a Java virtual machine, and it doesn't run as root or do setuid root. It can enable the user to perform a single task; it can provide an organized collection of tasks (with a built-in search mechanism); and it can include GUIs for monitoring the system.
- **Server-side daemon and command-line interfaces.** These can be written in C++ so Java doesn't have to run on the server being administered.

Communications between the client and server are secure, non-blocking, and transparent to the application. Encryption is supported (pending export compliance approval), as are security plugins.

Rhino features

Rhino has so many features, we had to list them on a [separate page](#). If you want to know more about the capabilities of Rhino, take a look.

How to learn more about Rhino

It's best to begin with the [introduction to basic Rhino concepts](#), which includes links to more in-depth discussion of each topic. You should also look over the descriptions and screen shots of the most important [GUI components provided by Rhino](#).

Throughout your reading, you may find it helpful to refer to the Rhino API documentation: the [package index](#), the [class hierarchy](#), or the [index of all fields and methods](#) (big!).

When you're ready to get started, jump into one of the topics below.

If you want to create a new Task and add it to an existing TaskManager...

Note that your new Task doesn't *have* to be plugged into a TaskManager; it can stand alone. Either way, you will want to read the [How to Write a Task](#) document. Then see the [How To Customize the Task Manager](#) document for instructions on plugging in your new Task.

If you want to create a new application...

[mkrhinoism](#) is all you need to know; based on your input, it will generate a complete sample ISM which you can build and install, and then use as a starting point for your development work. *Err... actually,*

mkrhinoism hasn't been modified since Rhino was converted to use autoconf & automake, so it's not actually usable at the moment. You probably want to ignore this paragraph for now.

If you'd prefer a more in-depth answer, including notes on analysis and design, see the [How to Write a Rhino Application](#) document.

Feedback

Please contact [**rhino@oss.sgi.com**](mailto:rhino@oss.sgi.com) if you have any comments or questions regarding these documents or the Rhino APIs. Your feedback will help us improve Rhino and its documentation.

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

Basic Rhino Concepts

This is an introduction to basic concepts which will be used throughout the rest of the Rhino documentation.

Task

A **Task** is defined as *an atomic operation that changes the state of the system*. For example, Tasks that deal with user accounts might include "Add a User Account," "Modify a User Account," and "Change a User Account Password."

Each Task can be presented to the user through one (or both) of two interfaces: a terse, single-page **Form** or a more verbose, multi-page **Guide**. If both interfaces are defined for a Task, each will display a control that allows the user to switch between the two at any point while performing the Task.

Tasks can be launched from the command line and from many places in an application or applet which uses a **Task Manager**, a GUI component which contains organized sets of Tasks. The execution of a Task invokes one or more privileged commands (described below) on the server.

Multiple Tasks may be combined into a **MetaTask**, which is a GUI component containing a sequence of steps leading to a high-level goal.

Item and Category

Item and Category are the mechanism by which the server tells the client about the state of the system.

An **Item** represents a physical or logical entity that is manipulated by system administration operations. Each Item has an associated type and a unique name within that type. For example, a user account named "foo" can be represented as an Item of type "user account" and unique name "foo".

A **Category** represents a dynamic collection of Items of a specific type. For example, the collection of user account Items can be represented as a Category.

Privileged Command

A privileged command (or "**priv command**" for short) is a command-line program which is run on the server to change the state of the system. Priv commands are not setuid, but they are run by the `runpriv` command, which is setuid. `runpriv` checks privileges, makes a log entry, and runs the priv command as the user who invoked the command.

Priv commands have been in use since IRIX 6.3.

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

GUI Components

Rhino provides some high-level GUI elements for displaying Tasks, collections of tasks, the status and relationships of Items operated on by those Tasks, and the results of those Tasks.

The following screen shots have been scaled to 50% of their actual size, and are links to full-size images. These examples are taken from the FailSafe 2.0 Cluster Manager GUI.

Windows:

- [Form](#)
- [Guide](#)
- [ResultView](#)
- [TaskManager](#)
- [MetaTask](#)

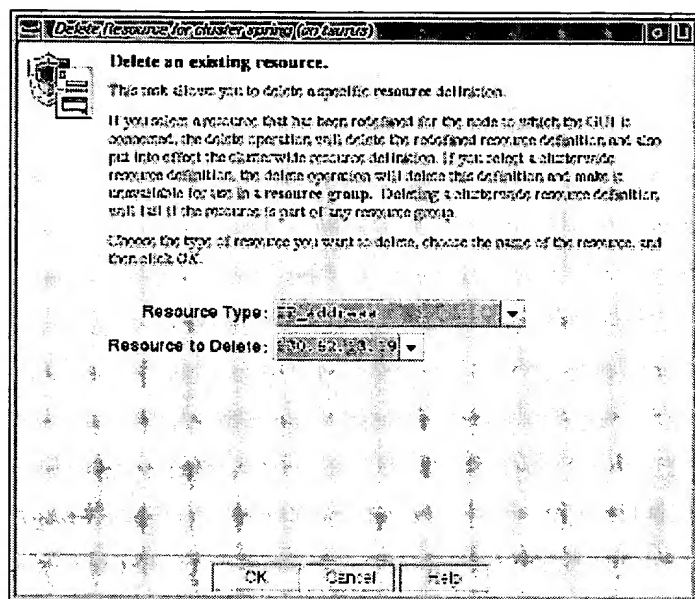
Sub-components:

- [RichText](#)
- [TaskShelf](#)
- [ItemView](#)
- [ItemTable](#)
- [TreeView](#)

Form

A **Form** is a single-page GUI for performing a Task. It contains a product-specific Task icon in the upper left corner, the Task title, and some introductory text describing the inputs the user is expected to type or choose. The Task-specific inputs themselves appear in the middle, with generic OK/Cancel/Help buttons at the bottom. All text in blue behaves like a hyperlink which launches glossary information in a separate small window.

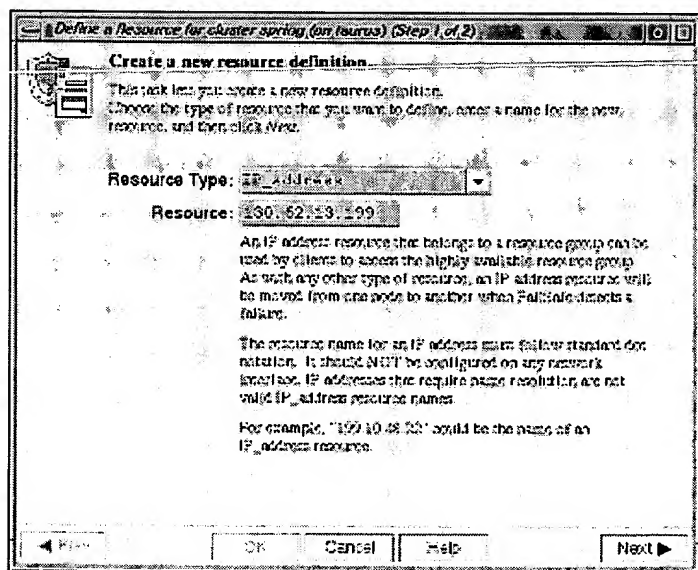
The purpose of the Form interface is to make the entry of Task parameters simple and fast. It is suitable for Tasks of low complexity and a small number of parameters. Forms are the preferred interface when the typical users are knowledgeable and comfortable with the system being administered.

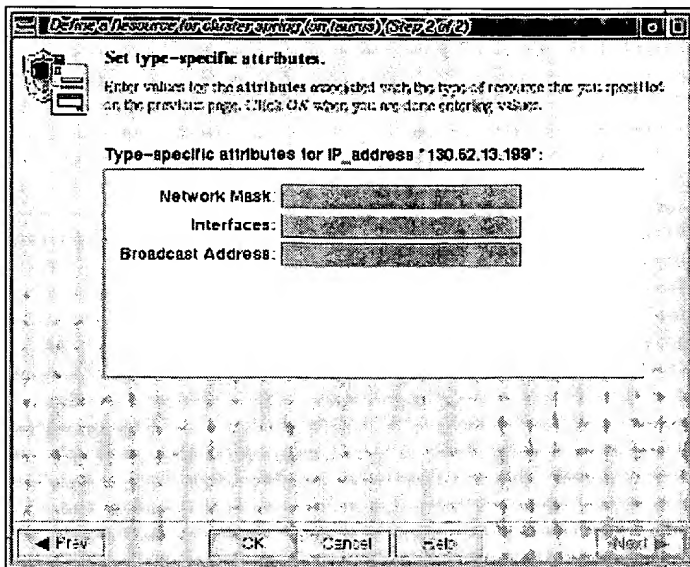


Guide

A **Guide** is a multi-page GUI containing explanatory text with a small set of labelled input components on each page. Like the Form, each page of the Guide contains a product-specific Task icon in the upper left corner, a title, and text describing the input expected of the user. The Task-specific inputs themselves appear in the middle, with generic Previous/Next/Cancel/Help buttons at the bottom. An "OK" button is presented when the user has navigated to the last page in the Guide. As in the Form, all text in blue behaves like a hyperlink which launches glossary information in a separate small window.

The purpose of the Guide interface is to provide step-by-step guidance on completing a complex task or a task with a large number of parameters. Guides are the preferred interface when the typical users are novices or not comfortable with the system being administered.

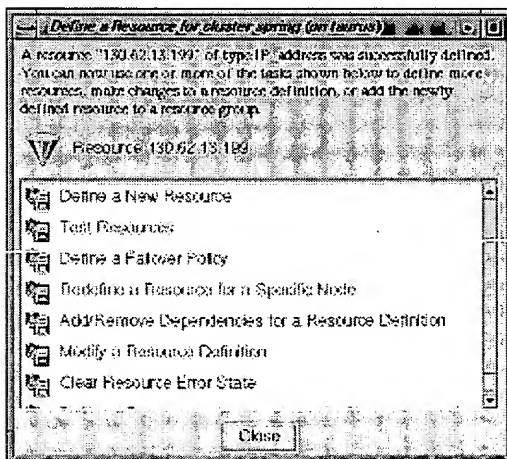




ResultView

ResultView is a window which displays the results of a Task which has been successfully completed. (If the Task could not be performed, the user is given an error message describing the problem, and the Form or Guide remains open until the Task is successfully completed or explicitly cancelled.)

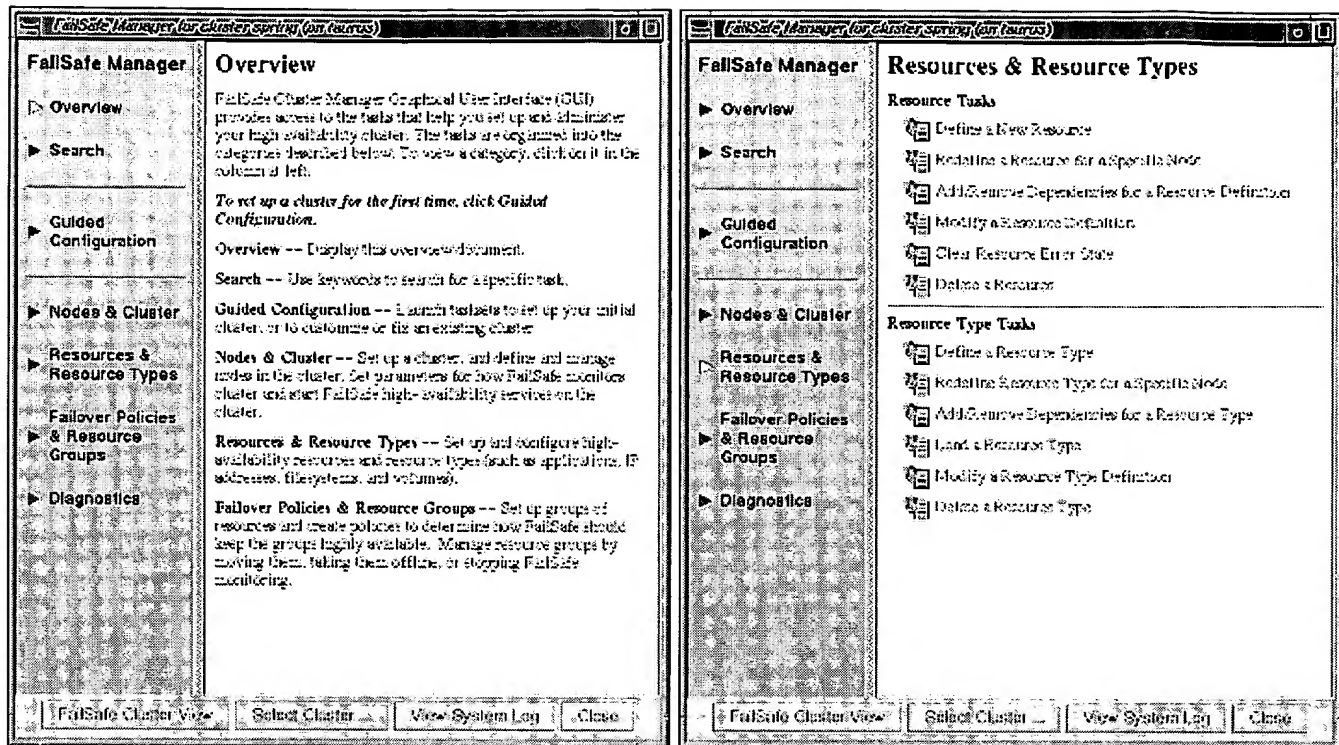
A ResultView contains a descriptive message, an icon representing the Item which was operated on (if applicable), and a TaskShelf showing the related Tasks which the user may want to launch next.



TaskManager

Task Manager is a front-end window that organizes and lists a product's Tasks for easy access.

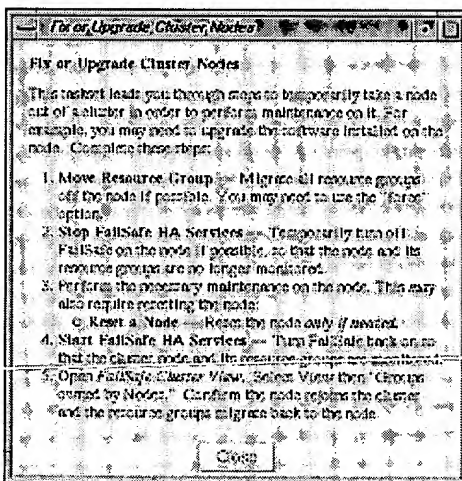
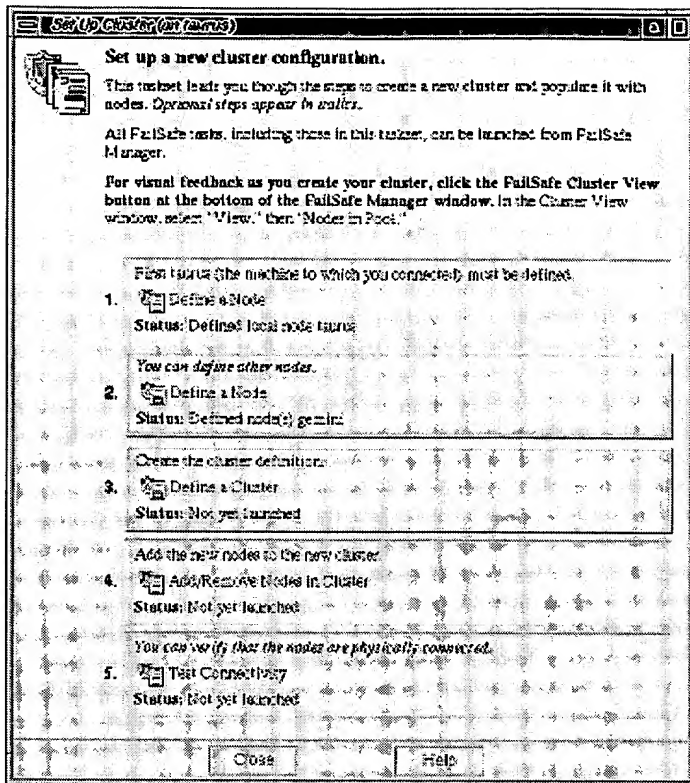
The Task Manager groups the product's Tasks into pages based on the types of Items that the Tasks operate upon. The list of pages appears on the left side as a table of contents. The Overview and Search pages appear in all Rhino applications, but the content of the Overview page is application-specific. For the FailSafe 2.0 GUI, all metatasks are grouped into a sixth page called Guided Help.



Related: TaskManager API page, screen shot, [How To Customize the TaskManager](#)

MetaTask

A MetaTask is a sequence of tasks which guides the user in performing a higher-level operation. There are two kinds of MetaTasks: "Smart" MetaTasks and ...the other kind. (This section is not complete.)



RichText

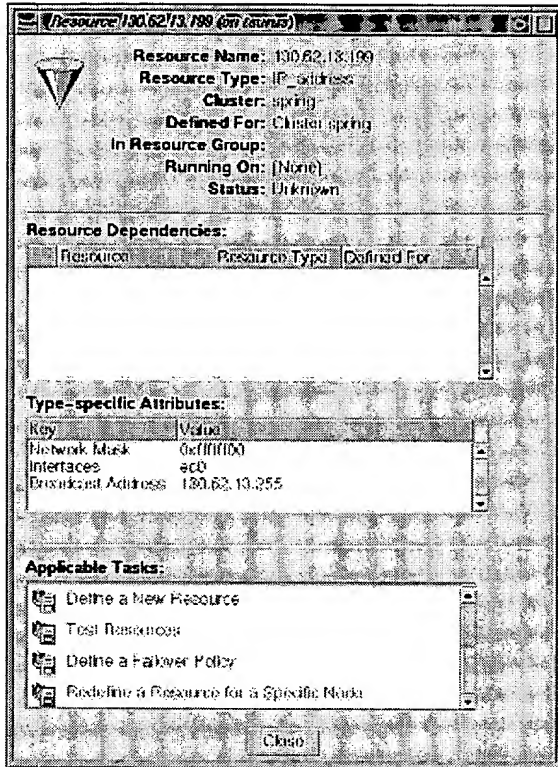
RichText is a text component that can display a small subset of HTML, including links. These links are most often used to bring up glossary definitions in another small window, but they can also be used to launch Tasks. Most of the Rhino components on this page contain one or more RichText components.

TaskShelf

A **TaskShelf** is a list of Tasks relevant to whatever GUI component contains the TaskShelf. The User can launch one of those Tasks by clicking on the Task name or icon. The TaskShelf is often dynamic, which means it will update the list of Tasks based on the state of the system.

ItemView

An ItemView is a window displaying all relevant information about an Item. The ItemView window displays simple key-value pairs at the top, application-specific contents in ItemTables in the middle, and a TaskShelf at the bottom. The Item's icon is shown at the top left, with the icon color indicating the Item's state.



Related: [ItemView API page](#), screen shot, [How to Write an ItemView](#)

ItemTable

An ItemTable shows information about all Items in a Category.

Related: [How to Write an ItemTable](#)

TreeView

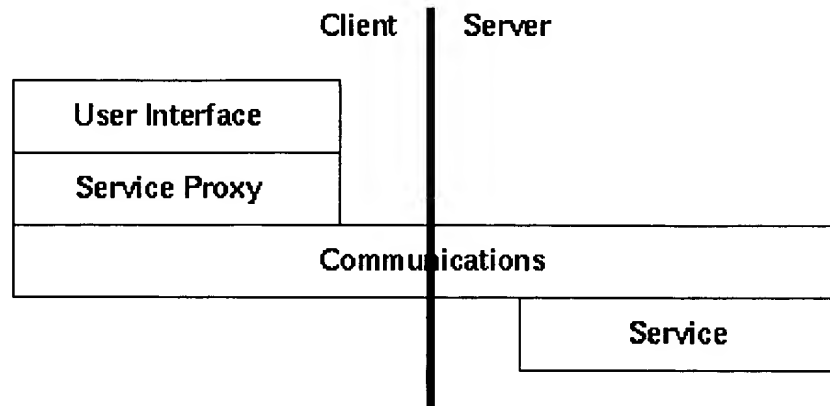
A TreeView displays Items that have a natural hierarchical relationship in an outline-style indented overview. Because it lets the user monitor the states of several Items at once, TreeView can be appropriate in a front-end monitoring window.

Related: [The Rhino TreeViewPane Component](#)

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

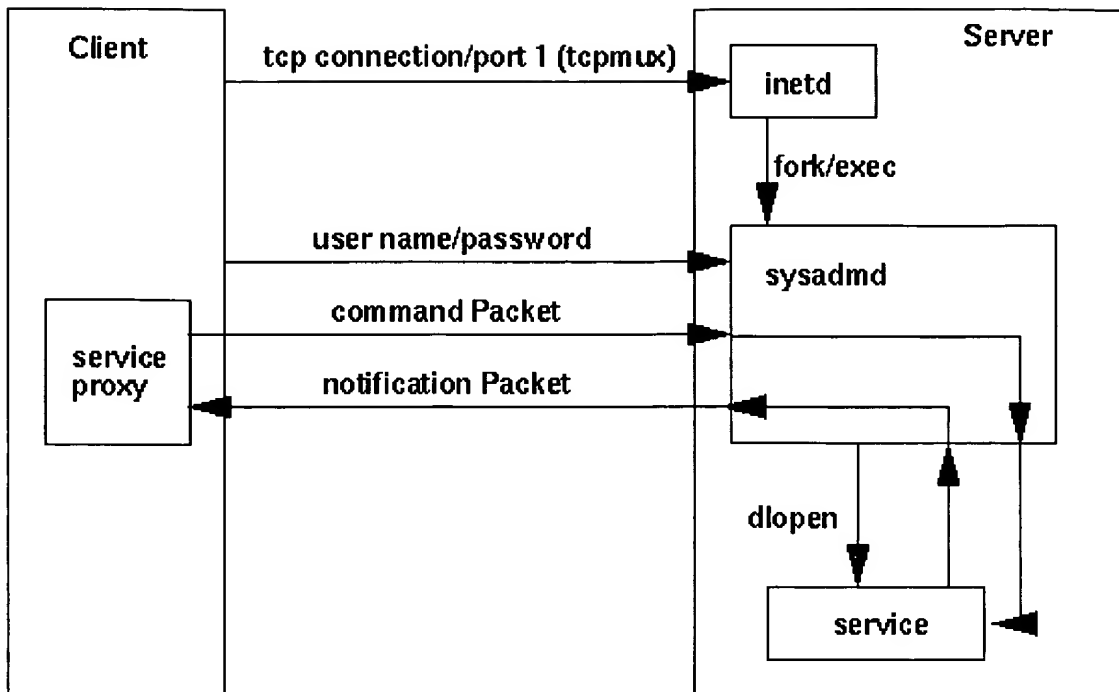
Rhino Architecture



The Rhino Architecture consists of three different interacting subsystems. The Communications subsystem handles the transfer of data between client and server. The Services subsystem models the system to clients and provides a mechanism for making changes to the system. The Services subsystem is further divided into client (or service proxy) and server components. The User Interface subsystem provides a framework and components for developing System Administration user interfaces.

Communications

Note: The material in this section is provided so that developers can understand how the Rhino architecture works. Rhino developers should never have to interact explicitly with the Communications subsystem; instead, Rhino developers use the services described in the next section.



The Communications subsystem is responsible for transferring data back and forth between the client and the server.

The Rhino server is called `sysadm`, and is typically started by the client via `inetd`. The client connects to port 1 on the server machine, which is serviced by `inetd`. The client makes a request to `inetd` that it start the "sgi_sysadm" tcpmux service, and `inetd` runs `sysadm`.

In order for the client to do anything useful, it must first authenticate itself with the server. When `sysadm` is started from `inetd`, the user must provide a valid user name/password combination. `sysadm` will not respond to any requests other than authentication requests until a valid user name/password has been supplied.

When `sysadm` is started by `inetd`, it is running as root. Once a valid user name/password is specified, `sysadm` sets its user id and its group id so that it is running with the permissions of the user name that was specified. In this way, the system is protected from security problems with `sysadm` because the user can't do anything via `sysadm` that he or she could not do by logging into the system.

Once the user has been authenticated, communication between the client and the server takes the form of commands from the client to the server and notifications from the server to the client. The basic unit of communication is the Packet, and each Packet contains a **type** which identifies which service it is associated with and a **selector** which indicates which command or notification is being sent. Additionally, each Packet contains key/value pairs of information which specify any additional information needed to convey the command or notification.

`sysadm` starts the **sysadm** service at startup. The **sysadm** service has commands that the client uses to load and unload other services. The client specifies in a Packet which service to be loaded, and `sysadm` looks in `/usr/sysadm/services` to find the dynamic shared object (DSO) which implements the requested service. `sysadm` dynamically loads the service, and henceforth any packets received by

`sysadmd` having the **type** for that service get routed to the service's **handlePacket** method. Additionally, a service may send Packets back to a proxy running on the client. The client matches the **type** of a Packet from the server to the appropriate proxy and calls its **handlePacket** method.

Services

The Rhino Architecture provides four services that clients can use to access the server system. All services are available via the `HostContext` accessor methods `getCategory`, `getAssociation`, `getTaskRegistry`, and `getPrivBroker`. A `HostContext` instance is initially available to clients that have implemented `RApp` or `RApplet` subclasses, and is typically accessible in other contexts from a `UIContext` instance. Callers of `HostContext` methods do not need to be concerned with sending Packets or loading services. The `HostContext` accessor methods return service proxies which encapsulate all interactions with `sysadmd`.

Category Service

The Category service is used by clients to get information about the system. On the server side, a Category monitors some aspect of the system, and maintains an Item for each entity. The client is notified when Items are added, changed, or removed.

The Category Service is described in more detail in Item and Category in Rhino.

Association Service

The Association service maintains state representing relationships between Items on the system.

The Association Service is described in more detail in Item and Category in Rhino.

Task Registry Service

The Task Registry Service fetches lists of tasks from the server based on a variety of criteria.

Privilege Broker Service

The Privilege Broker Service lets the client run privileged commands on the server. This is the only way in which a Rhino client can make changes to the system.

`PrivBroker` provides a variety of ways in which the arguments may be specified to a privileged command. The `runPriv` method which passes arguments in the form of an `AttrBundle` is very convenient when the privileged command on the server uses `libsysadmParam` (see `/usr/include/sysadm/SaParam.h`) to parse its command line arguments. The Privilege Broker service translates the `AttrBundle` into a format that is compatible with the parsing done by `libsysadmParam`. Since TaskData (see User Interface section below) is derived from `AttrBundle`, it is possible that the `TaskData` containing the parameters that the user has specified for a Task may be passed directly to the `PrivBroker` service.

See the `runpriv(1M)` man page for more information on the Irix privilege mechanism.

User Interface

The user interface subsystem consists of a few high-level framework components, along with many smaller components which can be used to build applications. The high-level framework components are:

- Task provides a user interface for making a change to the system.
- Task Manager organizes tasks into pages from which the user can launch them.
- Item View displays information about an administered item.
- RApp is the base class used for deriving new Rhino applications.
- RApplet is the base class used for deriving new Rhino applets.

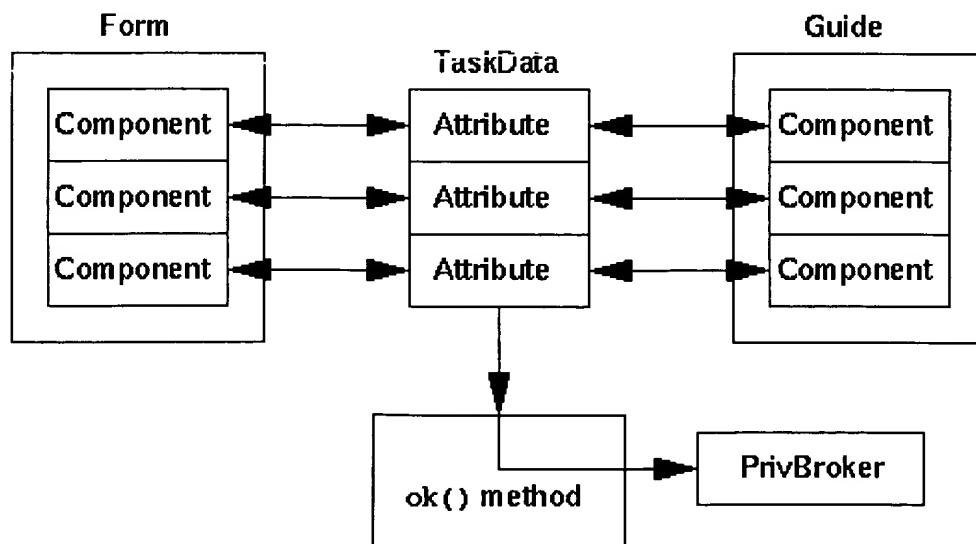
The other components include:

- ItemFinder is a JComboBox populated with Items from a Category or Association.
- ItemTable is a JTable populated with Items from a Category.
- EditableList provides a user interface for editing a list of entries.
- RichTextComponent displays formatted text.
- RichTextArea is a subclass of RichTextComponent that supports the display of glossary entries.
- RCheckBox, RButtonGroup, RDialog, RFrame, RLabel, RPanel, RPasswordField, RRadioButton, and RTextField are Rhino specializations of similarly named "J" Components from Swing.

The DynamicSize and DynamicSizeLayoutManager interfaces are the basis for Rhino dynamic geometry management. Rhino dynamic geometry management is implemented by Components whose heights depends on their widths, such as RichTextComponent. Dynamically sized Components implement the DynamicSize interface, which DynamicSizeLayoutManagers can use to determine the correct height to allocate for a Component given its width.

See the com.sgi.sysadm.ui, com.sgi.sysadm.ui.richText, com.sgi.sysadm.ui.event, and com.sgi.sysadm.ui.manager packages for complete listings of Rhino UI Components.

Task Architecture

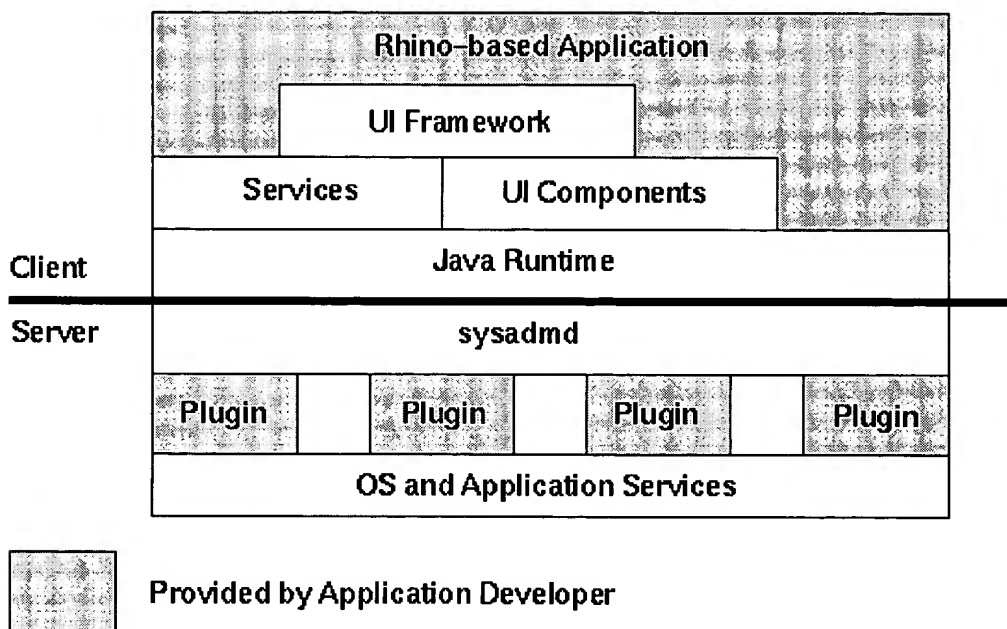


A Task provides one or more user interfaces which prompt the user for parameters for making a change to the system, and an *ok* method that gets called when the user presses the OK button. Since a Task can

have more than one user interface (Form and Guide), and since the user can switch back and forth between user interfaces, the TaskData mechanism is provided so that data is not lost when the user switches user interfaces.

The Task's internal representation of what the user has entered is stored as Attributes in the TaskData. Each Component in each of the user interfaces of a task is bound to an Attribute in the TaskData, so that when the Component changes, the TaskData is changed, and when the TaskData changes, the Component is changed. Thus, all input is preserved when the user switches back and forth between Form and Guide, and the *ok* method can get the parameters to pass to the Privilege Broker Service from the common TaskData rather than querying the user interface Components.

Architecture of a Rhino-based Application



The above diagram illustrates how the various pieces of the Rhino Architecture fit together with an application. On the client side, the application is in control, and uses the Rhino infrastructure to help implement its functionality. On the server side, `sysadmd` is in control, and it accesses developer-supplied plugins as requested by the client. A Plugin can take the form of a Category, Association, or Privileged Command, and it can also take the form of Task Registry entries.

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

How to Write a Rhino Application

The following steps are meant to be broad guidelines; your particular application may require more or fewer steps than those listed below.

1. **Analyze the UI requirements of your application** in terms of the Rhino UI classes. Understand the Rhino users model.
 - o [GUI Design and Implementation](#)
 - [Analyzing UI Requirements](#)
 - o [Rhino Classes](#)
 - o [Rhino Users Model](#)
2. **Analyze server-side requirements.** Identify the Category(s) that need to be administered. Identify Item attributes and Category attributes (if any).
3. **Generate example ISM.** Use [mkrhinoism](#) to generate a self-building example ISM that will help you get started using Rhino to create your application.
4. **Provide an API** that allows Rhino to determine the Item(s) belonging to the Category(s) and provide notification of changes to the state of the Item(s).
5. **Implement the server-side functionality:**
 - o [Priv Commands](#)
 - o [Item and Category](#)
 - o [Association](#)
6. **Implement the client-side user interfaces:**
 - o [Item and Category](#)
 - o [Associations](#)
 - o [Tasks](#)
 - o [Icon Renderers](#)
 - o [Name Renderers](#)
 - o [ItemViews](#)
 - o [ItemTables](#)
 - o [TreeViewPanels](#)
 - o [Task Manager](#)

GUI Design and Implementation

This document describes the overall design and implementation process in developing the client GUI side of a Rhino-based system administration product. This document focuses on the client GUI; the separate Rhino Tutorials document describes the process of developing the complete Rhino-based product, both client (front-end) and server (back-end). For example, the "Implement GUIs" section here briefly discusses only implementation time estimates, whereas the "Implement the client-side interfaces" section in the Rhino Tutorials page provides specific GUI components that you may want to use during implementation.

Note: Although presented as a sequence of orderly steps, this process is **highly iterative** in actual practice!

- I. Keep in mind...
 - II. Understand the problem space
 - III. Develop functional specification for setup and modification
 - IV. Develop functional specification for monitoring
 - V. Do mockups and design review
 - VI. Implement GUIs
 - VII. At alpha, conduct usability study(s)
-

I. Keep in mind...

Keep in mind these common systems administration requirements:

- Initial configuration and setup
- Production-mode modification
- Monitoring

II. Understand the problem space

1. Interview at least 2 or 3 customers, and do workflow analysis on what you've observed and on the war stories they tell you
2. Check out competitors products (install and evaluate, read brochures on the web, get info from customers, etc.)
3. Talk with internal SGI marketing, SEs

III. Develop functional specification for setup and modification

Based on user's needs, develop functional specification for initial configuration and setup, and for production-mode modification.

1. Based on data collected to understand problem space, identify:
 - o The types of objects to be administered (ex., user, filesystem)
 - o The system administration operations to be performed on those objects (ex., "Add User", "Mount Filesystem")

2. For each operation, identify the following. (These directly influence the design of the GUI. They identify the text, inputs, and outputs for the GUI. For an example functional specification that uses these fields, see the [FailSafe 2.0 GUI functional specification](#).)
 - o User's goal: what the user hopes to accomplish
 - o Context: how this goal fits into user's larger goal
 - o Prerequisites: validations the GUI can pre-calculate
 - o Up-front info: info the GUI knows that the user cannot directly change in this operation but that can help the user avoid an error in this task, also ramifications of doing this task (what will happen)
 - o User fills in: input required from user that GUI cannot predict
 - o User chooses: input required from user that GUI can predict (from set of predefined choices)
 - o Ordering: I/O sequence between GUI and user
 - o Possible results: possible success or failures for the operation
 - o CLIs used: command-line interfaces that the GUI will use to perform this operation
3. Organize the operations using customer data (this will directly influence the design of your front-end GUI)
 - o Organize operations into categories (ex., "User/Group Management")
 - o Organize operations into high-level goals (ex., "Set Up a New Cluster")

IV. Develop functional specification for monitoring

Based on user's needs, develop functional specification for monitoring GUI.

- Identify the object states that are meaningful and important to the user (ex., cluster node is down)
- Identify the relevant relationships between objects (ex., resource groups run on a cluster node, resources are of a particular resource type)
- Based on user data collected, identify constraints on the front-end monitoring GUI (ex., small screen real estate, overview of entire set of objects in their various states)

V. Do mockups and design review

- It can be useful to do a mockup or two first, to get an idea whether you're planning for too much text in the window or whether a particular Task should be implemented as a single-page Form or a multi-page Guide. For actual examples, see the [FailSafe 2.0 GUI Design Review](#) slides (Showcase). The mockups should derive easily from your functional specification.
- After putting together mockups, you might want to hold a design review. Members of the [Rhino team](#) are willing to attend and consult with you.

VI. Implement GUIs

Based on the functional specifications, implement GUIs for initial configuration and setup, production-mode modification, and monitoring.

- GUI for one operation can take 1-5 days to implement.
- GUI for monitoring can take several weeks to implement.

VII. At alpha, conduct usability study(s)

When you release your first alpha, it's a good time to conduct usability study(s).

- Ask neophyte user to accomplish some of the mainstream high-level goals (ex., "Set up a new cluster"), and evaluate where and why they get lost along the way
- Based on this data, revise and improve GUI

Analyzing UI Requirements

Before you can implement a Rhino-based application, it's important to understand what the user needs to achieve with your product, and how they're likely to use it. We've found it useful to talk to target customers and to SGI marketing and to check out competitors' solutions. Consultant Richard Anderson was instrumental in developing the following customer workflow analysis techniques.

Talk to Customers and Marketing

Customers can provide accurate, unpredictable information about product requirements, if you know how to ask them questions. Ask them to tell you stories about their most recent success and most recent failure using existing software to attempt to solve problems that your product is intended to solve. It can also be useful to talk with SGI marketing, who can be very knowledgeable about customers and the problem space.

Logistics

Make sure interviewees know generally what to expect. Tell people what will happen at a high level, but don't show them questions beforehand because that can lead people to start to take on other peoples' roles to try to answer the questions and their answers tend to become inaccurate, shallow, or vague. People can feel like they're a representative for others. If that happens, then they'll tell you what they *think* other people do, which can be inaccurate. Encourage them to talk about stories from their real lives.

Ensure up front that you say you're *not* there to evaluate what they do. Also state that you're *not* there to find out what they need so you can help them right then.

We've found it useful to interview as few as three different customers. Others say they interview around 15 people, no matter the size of the domain. There turns out to be a lot of commonality in how people do things.

We take one or two interviewers to interview one person. Don't fill the room with interviewers. More than about three questioners against one interviewee can be semi-ominous and make the interviewee clam up. Shoving everyone in a conference room can remove a lot of useful contextual information. You get better data if you can go to customers' offices and check out their surroundings and see what other tools they use besides yours and how they use them. However, good data can also be gotten through telephone interviews.

Take good notes. Be interested in everything they say (not just some things they say). Refrain from being judgmental (even a raised eyebrow). Don't take the role of an expert. Interviewees should take that role; you are learning from them.

Audiotape or videotape might work in some situations. Take a tape recorder, and ask if you can tape them. Later you can transcribe the recording and refer back to the transcript to make design decisions.

The interview should last an hour or two. Make the interview feel like a quality experience for the interviewee. Note that what they've said has been helpful. Afterwards, thank them and give them some token of appreciation for their time.

What to ask

Sometimes the question you want an answer to isn't the one you want to ask directly, because it won't get an accurate answer. Such questions relate to how a person does their work, when or how often they do something, why they do or don't do things. The reason these questions tend not to elicit accurate answers is that most of the time people don't watch themselves as they work; they just do their work. People don't usually take detailed notes as they work, so typically it's difficult for people to answer meta-questions accurately.

So what questions do you ask? The answer to this can come from focusing on context. If you go into the context and observe, you can ask questions as what's happening is actually happening. Your presence does affect things, but not as much as you'd expect. Context is key. People's thoughts and memories are different when they're in context.

What if you can't go to the customers' context (workplace)? If you bring them to you at SGI, you can try to restore the context:

- Have them bring pieces of the context with them. This can include files, documents, stuff they use and create, people they work with. Those pieces will help them remember context and other pieces of context that they *didn't* bring. Also, sometimes you can keep those pieces for your own reference and analysis later.
- Get people to tell you stories about work events that actually happened. These stories also contain context. Stories can trigger recall of other stories, and can help you ask questions (usually clarifications) that people can answer easily and accurately.

Here are some example questions you might want to ask customers. You may not want to explicitly use the questions in this list in an interview, but you can use the list as a reference to check if you've learned the answer to every question yet; if you haven't, then use the questions to provoke more discussion. These questions also may be too general; you should modify them as needed to focus on the product you're working on.

- **Initial context**
 - Get stuff off your chest.
 - Tell me about your job.
 - What does your group do?
 - Why did you bring an armadillo? (whatever piece of context they brought)
 - Describe your computer and your software.
 - Do you anticipate or dread coming to work? Why?
 - What is your role?
 - Who do you work with?
 - Where do you add most value?
 - Do you like your computer? Why (not)?
- **Information flow**
 - How is a new project started?
 - How do you depend on others?
 - How do you exchange information?
 - Where do you get your information?
 - Where are you when you need it?
 - How many information sources are there?
- **Projects overall**

- What are your projects like?
- Tell me about the project from hell.
- Tell me about the project you're most proud of.
- What was the hardest problem? How did you solve it?
- Who are your customers?
- How do you judge a project's success?
- How do you know a project is done?
- How do you deliver your products?
- **Day-to-day work**
 - What did you do yesterday?
 - What do you spend most of your time on?
 - Probe for wasteful activity.
 - What are your days like?
 - What would you delegate?
 - Tell me about that project you mentioned.
 - What makes your job hard?
- **Extract more info/Test our understanding**
 - Why did you do that?
 - What does that mean?
 - Is that what you expected? Why?
- **End**
 - Do you want to say anything else?
 - Thank them for useful info (Give reward: T-shirt, candy bar, gift certificate)

A different kind of question that you might use a lot is the clarification question. For example,

- What is that?
- What does that mean?
- Why did you do that?
- What's the purpose of that?
- Is that what you expected?
- What happened in that situation? What went on?

Check Competitors

Competitors typically have Web sites devoted to explaining their products, often with useful screen shots. It's worth taking a look at the features they offer, then assess that in light of what you've learned from talking with customers and marketing.

Rhino Classes

Rhino classes provide a means to represent the system to be administered and to perform user-specified operations to change system state:

- **Item** represents some system entity to be administered (ex., a cluster, an XLV volume, a filesystem). An Item usually has a unique icon in the GUI, reflecting its current state.
- **ItemView** is a window displaying relevant information about an Item.
- **Category** is a class of Items (ex., cluster nodes).
- **CategoryView** is a window displaying relevant information about a Category.
- **TreeView** displays Items that have a natural hierarchical relationship in an outline-style indented overview. Because it lets the user monitor the states of several Items at once, TreeView can be appropriate in a front-end monitoring window.
- **Task** is a window that users interact with. When the user clicks the OK button, the Task calls a CLI on the server to perform an atomic operation that changes the state of an Item. Tasks can be combined to achieve a high-level goal (a "metatask").
- **ResultView** is a window that appears when the user has completed a Task successfully, presenting a list of Tasks that the user may want to launch next.
- **Task Manager** is a front-end window that organizes and lists all of a product's Tasks, for easy access.

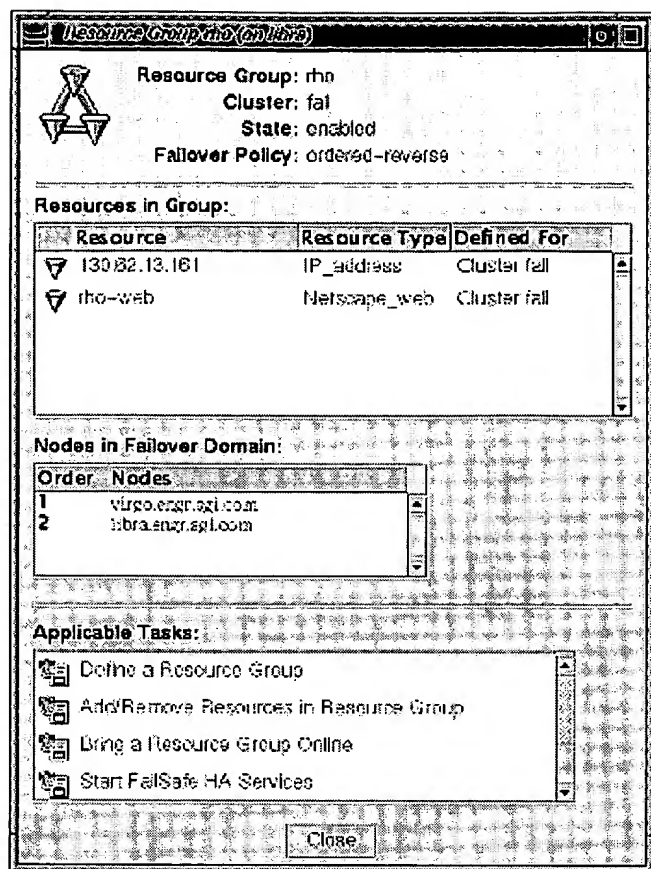
For all applications that the Rhino team has encountered, the above elements sufficed to satisfy administration requirements. If you encounter a requirement that is not satisfied by any of the above components, please [email the Rhino team](#). We await your feedback for additional requirements.

Screenshot Examples

The following screenshots have been scaled to 60% their actual size. These examples are taken from the FailSafe 2.0 GUI product which is based on Rhino.

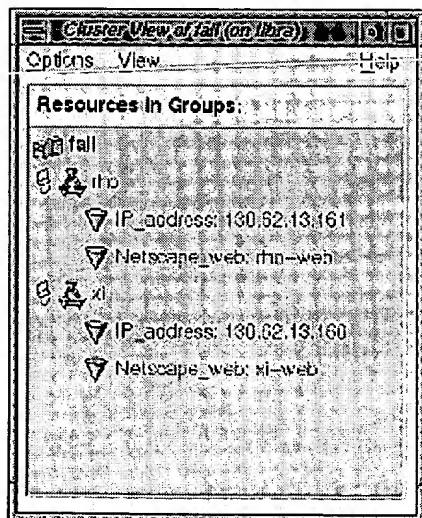
ItemView

The ItemView window displays simple key-value pairs at the top, application-specific contents in ItemTables in the middle, and a task shelf at the bottom. The Item's icon is shown at the top left, with the icon color indicating the Item's state.



TreeView

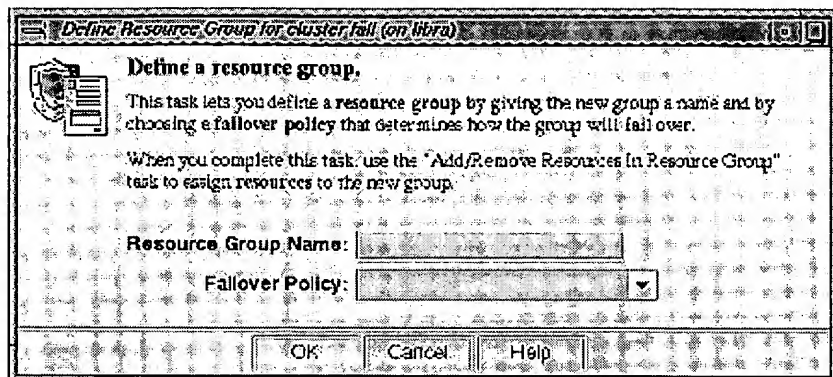
The TreeView shows Items that have a hierarchical relationship. In this example from FailSafe 2.0 GUI, three different kinds of Items are shown; the cluster "fal" contains two resource groups "rho" and "xi," and each resource group contains two resources.



Task

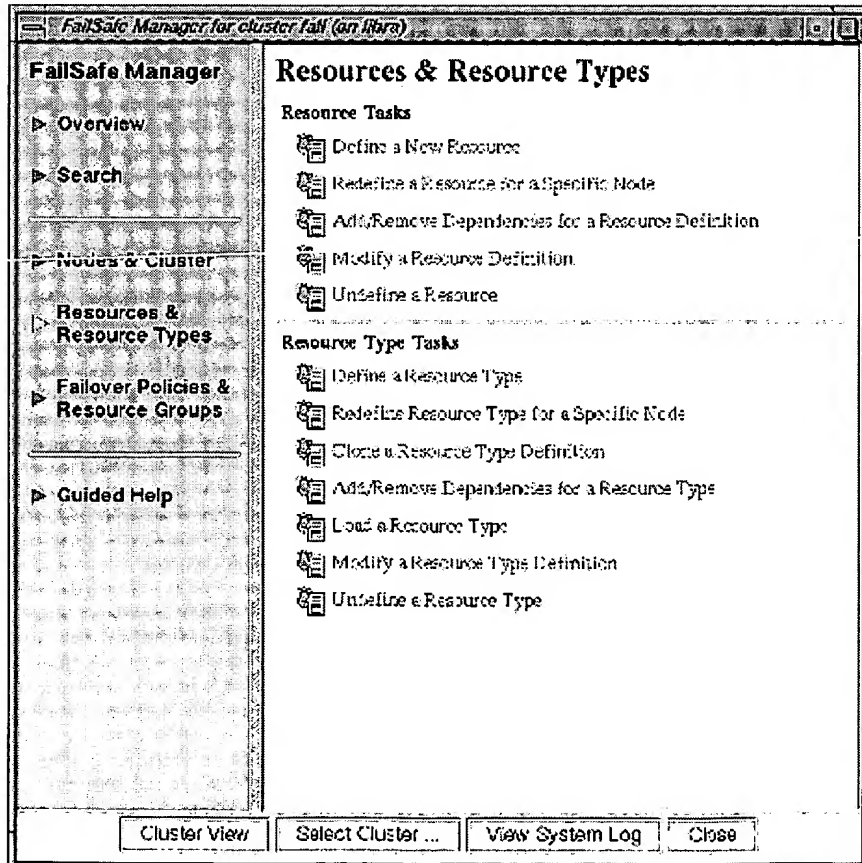
The Task window has a product-specific Task icon in the upper left corner (in this case, the FailSafe 2.0

GUI shield logo behind the generic Rhino Task logo). After the Task title, some introText follows, describing the inputs the user is expected to type or choose. Application-specific inputs themselves appear at the bottom. All text in blue behaves like a hyperlink and launches glossary information in a separate small window.



Task Manager

The Task Manager groups the product's Tasks into pages based on the different types of Items that the Tasks operate upon. The pages appear on the left side in the table of contents. The Overview and Search pages appear in all Rhino applications, but the text content on the Overview page is application-specific (describes the application and the application-specific categories). For the FailSafe 2.0 GUI, all metatasks are grouped into a sixth page called Guided Help, rather than presented in a special section at the top of each page.



Rhino Users Model

Rhino implements policies that maintain a consistent look and feel across all Rhino-based applications and that govern the end user's interaction with Rhino-based applications. These policies form the *Rhino users model*.

What do we mean by *users model*?

- What the user sees on the screen
- How the user interacts with the product
- What the user expects of the product
- How the user achieves a goal

What makes system administration difficult

This section explains where the Rhino users model came from.

The Rhino team identified possible end users for Rhino-based applications: Unix-illiterate bright professionals, such as animators and graphics artists, and systems administrators who seek tools to improve their efficiency. We interviewed these people using techniques described in Analyzing UI Requirements and thereby learned what makes system administration difficult:

- **Prerequisites and background knowledge.** Users get stuck when they must perform an administration task and don't know enough about the system to get started or to complete the task. For example, a user wants to share a file on the network, but doesn't know about `exportfs` and doesn't know that to share an individual file they must share the file's directory.
- **Information overload.** Users lose patience and feel overwhelmed when they must perform an administration task and are deluged with information from many sources: "Read this first!" documents packaged in the system box, man pages, release notes, books, the Web, and so on. Their frustration is compounded when the information is only partly relevant or consistent.
- **Risk of system damage.** Users hesitate before performing an administration task that they perceive may damage the computer, whether or not there is real risk. Other users try ad hoc methods and unknowingly do system damage when trying to perform an administration task. For example, a user accustomed to a single-user system like a Macintosh might always log into the `root` account; to free up disk space, the user might remove `/unix` among other large files.
- **Lack of confidence in result.** Users who try to perform an administration task are not certain that what they tried had the intended effect. For example, most commands issued at the command line return an invisible return code, and few state in plain English what has happened and what ramifications to expect.
- **Entry points difficult to find.** Users and systems administrators use different terminology when thinking about their systems, which makes it difficult for users to find and use the tools they need. For example, a user who wants to get a modem working might search for "modem" and not find that they must install `oe.sw.uucp`.
- **Error recovery.** When something goes wrong, users get stuck and don't know what to do to fix

the problem. For example, a user who unwittingly removes `/unix` might continue to use the system for days or weeks until the system is rebooted, at which point it can be difficult to track down what went wrong, why, and how to fix it.

- **Deciding what to do next.** Users don't know what to do next when they've run an unfamiliar command, whether the command seemed to have the desired effect or not. Users get confused when there is no feedback informing them how far they've come in an administration task or what their options are at any given point.

Rhino principles of interaction

Based on the above findings, the Rhino team developed the following principles of interaction:

- **Make prerequisites explicit and complete.** Telling the user up front what they will need to accomplish the task eliminates the need for them to seek this information elsewhere, thereby saving time, avoiding frustration, and building the user's trust in the application. This can eliminate the problem of having to know arcane prerequisites and mysterious background knowledge.
- **Hide unnecessary details by default.** Getting secondary concerns out of the way helps the user focus their attention on the task at hand, saving time and avoiding frustration. This can alleviate the information overload problem.
- **Always tell the truth about system status (if possible).** When the application gives accurate status, the user has confidence in the results and trusts the application to do the right thing.
- **Tell the user what will happen before they perform a task. Also tell the user what did happen after they've performed the task.** Giving users feedback before and after a task helps them understand the ramifications of performing the task, avoiding frustration and building the user's trust in the application. This can reduce the problem of error recovery.
- **Identify problems as soon as possible.** Users' time is precious. For example, an animator working on a *Jurassic Park* sequel may have a pressing deadline to finish designing a dinosaur model for the next morning's dailies. It would waste the user's time for the application to ask for additional input when the task has no hope of succeeding. For example, when the user tries to add a modem, the application should inform the user if `oe.sw.uucp` is missing and tell the user to install it.
- **Let the user know what can be done next.** Giving users alternative steps to take next helps them understand what they've done so far and how close they are to accomplishing their larger goal (they may be done), avoiding frustration, saving time, and building the user's trust in the application.

The Rhino users model in action

The Rhino principles of interaction have been embedded in the Rhino infrastructure:

- Tasks can be implemented as a Form, a no-frills window containing a set of inputs (suitable for expert users), and/or as a Guide, a user-friendly interface that distributes inputs across pages and includes text that gives guidance to help the user choose each input (suitable for novice users).

- At the top of the window, the Task introText explains prerequisites needed to perform the task.
- The Guide summary page lists all input changes that the user has made, and informs the user what will happen when they click the OK button.
- After the user clicks OK, the Task either fails and an error dialog appears to inform the user about the problem, or succeeds and a Rhino ResultView appears to inform the user what has happened and provide a list of Tasks that can be performed next.
- The Rhino application developer can bind validations to be performed when the user changes any input. When any one of these validations fails, the user can thus be immediately informed in an error dialog and given instructions and opportunity to fix the problem.
- Rhino ItemViews, ItemTables, and TreeView can display the system state. Rhino Items and Categories communicate dynamically to keep these views up to date, such that the Rhino-based application always tells the truth about system status.

Writing Priv Commands

- Introduction and Background
 - Environment of Priv Commands Run Via Runpriv
 - Dividing Functionality Among Priv Commands
 - Naming Priv Commands
 - Returning status from Priv Commands
 - Validating Input
 - Priv Commands Should be Atomic
 - Priv Commands are Logged
 - Passing Parameters to Priv Commands
-

Introduction and Background

As described by the Rhino Architecture document, all communication between the client GUI and the server is handled on the server end by a daemon called sysadmd. This daemon runs as the user who logged into the GUI, not necessarily as root. When a user wants to perform some administrative function that requires root access, a command must be run on the server as root. There are several components of the Rhino architecture which support this:

priv command

Also known as privileged command. A priv command is a command line program that requires root level access to run successfully. These commands reside in the `/usr/sysadm/privbin` directory, and are the commands that actually perform the changes to the system when the user runs a Task from the GUI. The priv commands are not setuid root, but must usually be run as root to be effective, since they perform operations that will fail if they are not running as root. There are two ways to run priv commands: via the runpriv command, or the root user can invoke them directly.

runpriv

A setuid program that takes the name of a priv command as an argument. It allows a non-root user to run a priv command as root if any of the following are true:

1. The user is running as root.
2. There is no root password on the system.
3. There is an file in the defaultPrivileges(4) directory granting the privilege to all users.
4. There is an entry in the privilegedUsers(4) database granting the user all privileges.
5. There is an entry in the privilege(4) database granting the user the requested privilege, and the user is not an NIS user.
6. The -auth auth-scheme arguments are provided, and the user passes the authentication test. If auth-scheme is unix, then the user must type the root password when prompted in order to pass.

Privilege Broker Service

One of the services provided by sysadmd. It allows the GUI to pass a request to run a priv command to the server. The Privilege Broker Service currently always uses the "unix" style of authentication. The GUI can use the runPriv family of Java methods (See the Task documentation for more info) to pass commands to the priv broker service.

The Privilege Broker Service is currently the only method for the GUI to run a command on the server. For some products, it may be desirable to have a more general way to run arbitrary commands on the

server. These commands would run as the user that logged into the GUI, not necessarily as root, and would therefore have many fewer restrictions. This general "command" service is not currently implemented, but could be written if it is deemed necessary.

Environment of Priv Commands Run Via Runpriv

There are several restrictions placed on priv commands by the runpriv (1M) program for security reasons:

- The priv command must be installed in /usr/sysadm/privbin
- The environment is cleansed of all but the most basic environment variables (see /var/sysadm/privenviron)
- The home directory is set to /var/sysadm/home
- The priv command runs with the effective uid set to 0 (root) and the real uid set to the uid of the user that logged into sysadmd.

This last restriction makes it impossible to use a script as a priv command, because by default IRIX systems are configured to refuse to run shell scripts if effective uid != real uid. In this case, the recommend solution is to write a C wrapper that sets the uid to the effective uid and then calls the script. For example:

```
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/wait.h>
#include <signal.h>

#ident "$Revision: 1.2 $"

#define SCRIPT "/usr/bin/script.pl"

/*
 * A C front end to a Perl script.
 * This is required so that runpriv (1M) can
 * execute the script.
 */
void main( int argc, char *argv[])
{
    int    status;
    /* Set the uid to the effective uid
     */
    if (setuid(geteuid()) < 0) {
        perror("setuid");
        exit(1);
    }
    status=execv(SCRIPT,argv);
    exit(status);
}
```

Dividing Functionality Among Priv Commands

▼ Rhino

[Overview](#)
[News](#)
[Documentation](#)
[FAQ](#)
[Mailing List](#)
[How To Contribute](#)
[To-Do List](#)
[Download](#)
[License](#)

▼ SGI Open Source

[Project List](#)

▼ SGI Support and Resources

[SGI and Linux](#)
[Supportfolio for Linux](#)
[Customer Services](#)

▼ Community Resources

[Newsgroups & Mailing Lists](#)
[Support Links](#)
[Open Source Sites](#)
[Linux Documentation Project](#)

Rhino news

Here's what's going on with Rhino, followed by a list of things that have happened (which is pretty at the moment). If you'd like to help, take a look at what we're doing.

The return of mkrhinoism!

Roger Chickering's **mkrhinoism** example code generator is back in Rhino version 1.3.5. **You** are minutes away from **mkrhinoism fun!**

1. Install `sysadm_base-mkrhinoism` (and its dependencies... all right, this part is less fun than other steps below.)
2. Run `mkrhinoism`.
3. `cd` to the directory where you told `mkrhinoism` to generate a customized example source and run `./makeme`.
4. Install the RPM packages you just built. (If you're not using RPM, never fear; just change "make rpm" part of the script to "make && make install".)
5. Run your new example program! If you chose "squidmgr" as your ISM abbreviation, your program is probably `/usr/local/bin/squidmgrtask` or `/usr/local/bin/squidmgrview`.
6. All right, this is the optional super-secret triple fun step. **Edit the code** and make it do something useful.

And in fact, here's a [screen capture](#) (26K) of several windows from the ever-popular **Beer Manag** thoughtfully generated by `mkrhinoism`.

If you have any trouble, send mail to [the mailing list](#).

Server-side API documentation!

We're now using `kdoc` to generate API documentation for the C/C++ code which runs on the server. The conversion has been pretty automatic (a couple of quick perl scripts to convert `//` comment comments to `/*` where appropriate), so we're not really taking much advantage of it yet, but you can that even without much effort, [the results](#) are pretty nice.

Hopefully Soon-To-Be-News

See the [TODO list](#) for a more complete list of known bugs to fix and new features to add, but here are a couple of the main things which we'd like to do soon (other than fixing bugs):

- **Put SSH support back in.** Rhino used to support SSH for communication between the client and server (so that you didn't have to send your password over an unencrypted connection), but had to remove that for export compliance. It would be nice to put this back. Also, rogerc has some good feedback on this.
- **Privilege manager/mkrhinoism tutorial.** Now that we have `mkrhinoism`, it would be fun to write a tutorial showing the steps from raw `mkrhinoism` output to something useful, like a **Privilege Manager**. It would be a pretty simple tutorial, and the end result would be a useful tool.

News

If you'd like this list of news to be longer, come help us make some.

- 2000/11/14 **Rhino 1.3.7** released. This just has a couple of minor new things to support the FailS GUI. The [ChangeLog](#).
- 2000/9/18 **Rhino 1.3.6** released! This just has a few bug fixes, but they're bug fixes that *matter*. [ChangeLog](#).
- 2000/9/10 **Rhino 1.3.5 released!** Hey, it's got more than just mkrhinoism; it's got a bunch of bug fixes & other improvements. Here's the long & boring [ChangeLog](#).
- 2000/8/18 Rhino-based FailSafe GUI demo'd at [Linux FailSafe Symposium](#). All goes well, and Rusty doesn't have to fall back on plan B (pouring a glass of water on his pants & run out of the building).
- 2000/8/8 [Linux FailSafe](#) released. It uses a Rhino-based GUI for administering high-availability clusters.
- 2000/8/7 **Rhino 1.3.4** released, and **these delightful pages** first appear on oss.sgi.com.
- 2000/7/13 **Rhino 1.3.3** released with the SystemImager GUI.

[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark](#) in

[products](#)[industries](#)[developers](#)[services & support](#)

5

ope

▼ **Rhino**[Overview](#)[News](#)[Documentation](#)[FAQ](#)[Mailing List](#)[How To Contribute](#)[To-Do List](#)[Download](#)[License](#)▼ **SGI Open Source**[Project List](#)▼ **SGI Support and Resources**[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)▼ **Community Resources**[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)

Rhino

URL: <http://oss.sgi.com/projects/rhino/>

Rhino is an infrastructure for building applications that configure, manage, and monitor hardware software. Rhino provides a common, consistent, task-based, internationalized graphical user interface (GUI), with built-in command-line interfaces (CLIs) that system administrators can use to write scripts. Rhino applications consist of two parts:

- **Client-side GUI in Java.** The GUI runs on any platform that has a Java virtual machine, a doesn't run as root or do setuid root. It can enable the user to perform a single task; it can provide an organized collection of tasks (with a built-in search mechanism); and it can include GUIs for monitoring the system.
- **Server-side daemon and command-line interfaces.** These can be written in C++ so Java doesn't have to run on the server being administered.

See the [FAQ](#) for more detailed information on Rhino, and [News](#) for information on what kind of work being done with Rhino. It's pretty big, and can benefit from a wide range of skills.

[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark in](#)

Changes for version 1.3.7:

Added `acconfig.h` to top-level `Makefile.am`'s `EXTRA_DIST` so that you can run `autoheader` etc. in the source tarball. (rusty)

Fixed bug which sometimes prevented nodes from appearing in the FailSafe Cluster View, caused by Swing 1.1.1 changes. (relph/wessmith/aseel)

Added a `ResourceStack.pushBundle()` method which takes a bundle name and its contents, and `ResourceStack.putString()`, for more control over the contents of `ResourceStacks`. (rusty)

Changes for version 1.3.6:

Fixed broken link in `doc/tutorials/Overview.html`. (rusty)

Fixed memory leaks & `realloc()` error in `StringFromFile`. (rusty)

Added `ResourceStack.pushBundle()` method which takes a bundle instead of a file name, for loading properties from (for example) an `Item`'s attributes. (rusty)

Moved `en_US.dude` files into `sysadm_base-client` & `-server` instead of having them in their own packages, as no one else seems to do that. (rusty)

Changes for version 1.3.5:

Fixed bug 796568 in `libsalog` where `runpriv` etc. would report the wrong error when there was no disk space left for the `salog` message. (rusty)

Fixed bug 798555, where a security exception was being thrown when the client was running as an applet & figuring out whether to place its own windows. (dlu)

Added test for `fam.h` to configure script. (rusty)

Irritating incorrectly-packaged test classes fixed. (rusty)

No longer running `javadoc` every time we build; now it only happens when `sysadm.jar` has been built. Also, `jar` files now depend on the `java` timestamp files. (rusty)

Images are now being included in `sysadm_base-dev-doc`. (rusty)

Clarified error message from `jarfiles` when we can't find a `fooP.jar` file for `foo.jar`. (rusty)

Added `@LDL@` to `libsysadmCategory_la_LIBADD` so that categories can link with it without having to explicitly link with `libdl` themselves. (rusty)

`mkrhinoism` now works with `autoconf` & `automake`! Its generated code also uses `gettext`; builds a `libtool` library & `rpm` packages; has `java` help, splash screens, desktop icons, `java` plugin support, and `GPL` headers. (rusty)

Added `/usr/jdk118/bin` to `JAVAPATH` (if `JAVAPATH` isn't set), so if you're using the IBM JDK (we are), you don't have to tell the configure script where to find it. (rusty)

Added configure test & `Makefile` rule for generating server-side API

documentation with kdoc (under doc/api/server), modified comments in C/C++ headers to make them visible to kdoc, & moved packages/webdocs/tutorials and packages/webdocs/api to doc/tutorials and doc/api/client. Because the server-side API documentation is only built if you have kdoc, the sysadm_base-dev-doc package now just includes whatever it can find.
(rusty)

Added missing "Provides:" lines to rpm spec file. (rusty)

Changes for version 1.3.4:

This is the first open-source version. Or, rather, sysadm_base-1.3.3 was the first open-source version, but 1.3.4 is the first version with this outstanding level of documentation.

Welcome to Rhino!

Rhino is an infrastructure for building applications that configure, manage, and monitor hardware and software. Rhino provides a common, consistent, task-based, localized, secure graphical user interface (GUI), with built-in command-line interfaces (CLIs) that system administrators can use to write scripts. Rhino applications consist of two parts:

- **Client-side GUI in Java.** The GUI runs on any platform that has a Java virtual machine, and it doesn't run as root or do setuid root. It can enable the user to perform a single task; it can provide an organized collection of tasks (with a built-in search mechanism); and it can include GUIs for monitoring the system.
- **Server-side daemon and command-line interfaces.** These can be written in C++ so Java doesn't have to run on the server being administered.

Communications between the client and server are secure, non-blocking, and transparent to the application. Encryption is supported (pending export compliance approval), as are security plugins.

Rhino features

Rhino has so many features, we had to list them on a [separate page](#). If you want to know more about the capabilities of Rhino, take a look.

How to learn more about Rhino

It's best to begin with the [introduction to basic Rhino concepts](#), which includes links to more in-depth discussion of each topic. You should also look over the descriptions and screen shots of the most important [GUI components provided by Rhino](#).

Throughout your reading, you may find it helpful to refer to the Rhino API documentation: the [package index](#), the [class hierarchy](#), or the [index of all fields and methods](#) (big!).

When you're ready to get started, jump into one of the topics below.

If you want to create a new Task and add it to an existing TaskManager...

Note that your new Task doesn't *have* to be plugged into a TaskManager; it can stand alone. Either way, you will want to read the [How to Write a Task](#) document. Then see the [How To Customize the Task Manager](#) document for instructions on plugging in your new Task.

If you want to create a new application...

[mkrhinoism](#) is all you need to know; based on your input, it will generate a complete sample ISM which you can build and install, and then use as a starting point for your development work. *Err... actually,*

mkrhinoism hasn't been modified since Rhino was converted to use autoconf & automake, so it's not actually usable at the moment. You probably want to ignore this paragraph for now.

If you'd prefer a more in-depth answer, including notes on analysis and design, see the [How to Write a Rhino Application](#) document.

Feedback

Please contact rhino@oss.sgi.com if you have any comments or questions regarding these documents or the Rhino APIs. Your feedback will help us improve Rhino and its documentation.

Rhino Application Programming Interface

User's Guide

How the API Is Organized

There are three levels to the API:

- All Packages
- All Classes (within a package)
- This Class (selected class).

Level 1 - All Packages

This level of the API provides links to the packages in the 1.0 release.

Level 2 - This Package

This level provides links to the classes and interfaces in a given package. There are three categories in the listing:

- Interfaces
- Classes
- Exceptions

Level 3 - This Class/Interface

This level begins with an index, followed by the detailed API. There are three categories at the class level.

- Variables
- Constructors
- Methods

A category is omitted when a class has no applicable entries.

Within these categories there is additional color coding as follows:

- Instance Variables
- Static Variables
- Constructors
- Instance Methods
- Static Methods

How to Locate Items

- To Browse A Package
 - Select a package from the list of All Packages. This list is the home page for the the Rhino API.
- To Locate a Class
 - Use the searchable index tool.
 - Or, select its package.
 - Select the class from the alphabetical index.
- To Browse a Class
 - Use the Next/Previous anchors to browse alphabetically.
 - Or, traverse the links within the class.
- To Locate a Method
 - Use the searchable index tool.
 - Or, scroll through the alphabetical class index to locate a method.

The Index

Each class/interface begins with an index of its variables, constructors and methods, sorted alphabetically. The entry consists of the declaration and short description. The description is the first sentence of the doc comment for that item. The index entries are linked to their corresponding entries in the application programming interface which immediately follows.

The Detailed API

The index is followed by the complete API for each entry. Within the three categories: Variables, Constructors, and Methods, the entries are presented in the order they appear in the source. This is done to preserve the logical groupings established by the programmer.

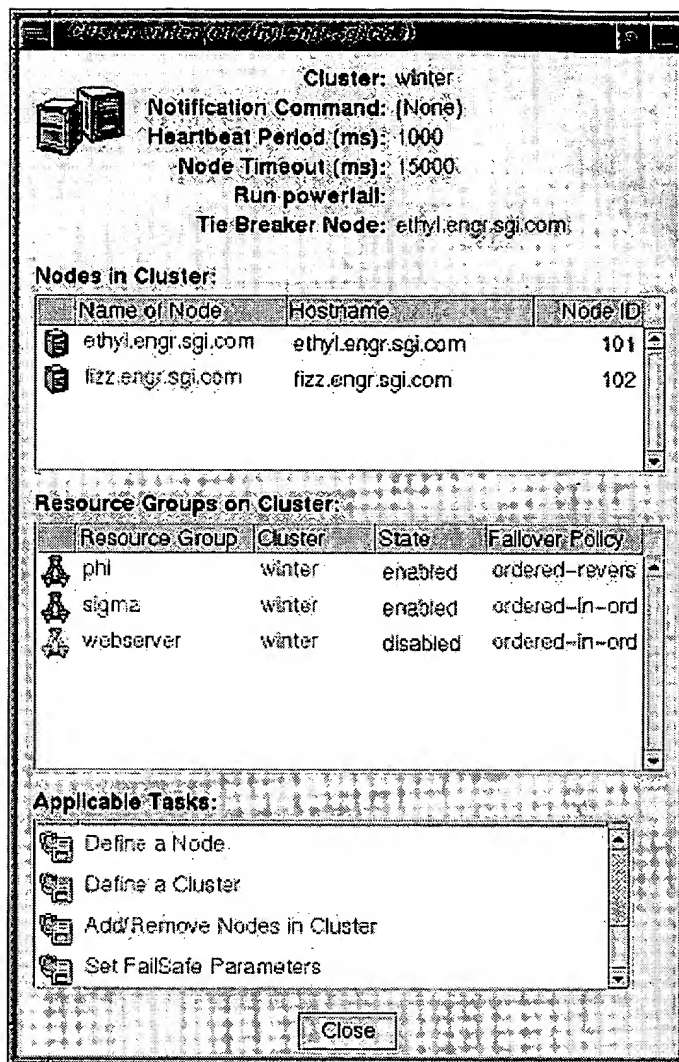
Where Are All the Links in the API?

- At the top of each class/interface there are navigational anchors to the other levels and to Previous and Next (class or interface).
- There are links in the class type of every method and variable definition.
- At the top of each class/interface there is a drawing of the tree structure down to the current class/interface, in which each superclass is a link.
- Every method contains a list of exceptions that it may throw. These are linked to the appropriate class.
- The superclass and interface references at the beginning of the class are links.
- Every See Also is a link.
- When a method overrides a method in the superclass, the API has the entry "Overrides: foo in class bar." Both foo (the method name) and bar (the class name) are links.

How to write a Rhino ItemView

Table of Contents

- [Introduction](#)
- [Overview of the ItemView's Sections](#)
- [Before you begin](#)
- [How to create an ItemView for a particular Category](#)
 - [No-Code ItemViews](#)
 - [Analyze Item's attributes](#)
- [Customizing the fields of the ItemView](#)
 - [The *field* properties](#)
 - [The *basedOn* properties](#)
 - [The *label* properties](#)
 - [The *method* properties](#)
 - [toString method](#)
 - [lookup method](#)
 - [richText method](#)
 - [renderer method](#)
 - [The *missing* properties](#)
- [Writing an ItemViewFieldRenderer](#)
- [Writing an ItemViewAdditionalInfoRenderer](#)
- [Controlling the Icon displayed on the ItemView](#)
- [Controlling the title of the ItemView](#)
- [Controlling the TaskShelf on the ItemView](#)
- [How to launch ItemViews](#)
- [Typical Resource File for an ItemView](#)



Introduction

This document is a reference for SGI software engineers who will be writing ItemViews for Rhino applications. An ItemView in the Rhino Architecture is a UI Component that displays all relevant information about a particular Item. The ItemView is the user's main source of information about the attributes of an Item, which include both static and dynamic information.

Overview of the ItemView's Sections

Shown below is a picture of an ItemView, with the different sections labeled.

Icon	Shown in the upper left corner of the ItemView. Typically, the Icon represents the type of Item being viewed, and additionally the state of the Item
Fields	Shown in the upper right corner of the ItemView. This section is divided into two columns, the left for the name of the field, and the right for the value of the field. The fields section is designed to show information about the Item that can be represented by fairly short Strings
Additional Info section	This section occupies the center of the ItemView. It is an optional section. This section is designed to show information about the Item that can't easily be represented as a single line of text. Examples include ItemTables, graphs, or additional icons. Any Java component can be shown here. If there are no components to show in this section, then the ItemView will not show the Additional Info section.
TaskShelf section	This section occupies the bottom of the ItemView. It shows a TaskShelf containing Tasks that can operate on the displayed Item in the Item's current state

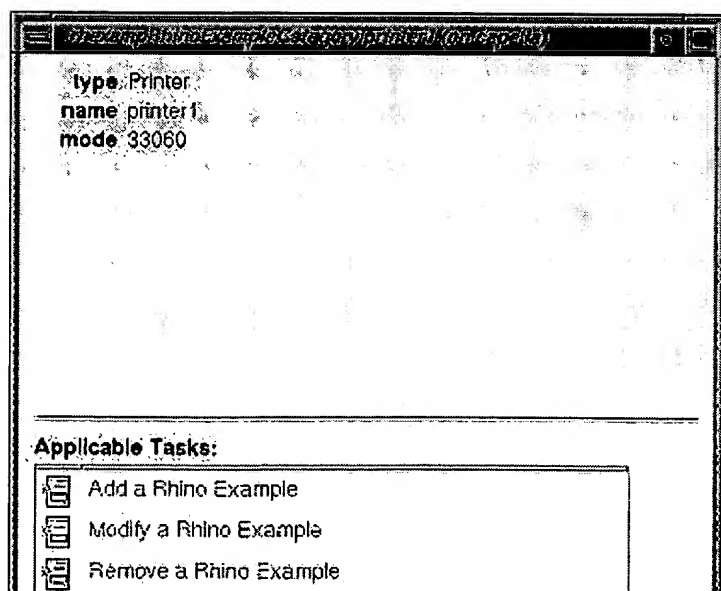
Before you begin

Before beginning to create an ItemView for a particular Category, it is necessary to understand the names and terms that the Rhino infrastructure uses in relation to Categories. See the [The Names of Categories on the Client and on the Server](#) documentation for more information.

How to create an ItemView for a particular Category

No-Code Item Views

While in the early stages of writing Categories, it may be desirable to show an ItemView that shows all of the Attributes of an Item. The ItemView supports this idea by means of a "no-code" ItemView. This version of an ItemView is not designed for use in a shipping Rhino application, but can be of great assistance while investigating the Rhino Infrastructure or for giving preliminary demos. No code or resource files need to be written to use the "no-code" ItemView - it can be launched as soon the server side Categories have been written and the Rhino infrastructure has been installed on the client. To turn the ItemView into a shippable ItemView, it is necessary to



provide resources that describe the way that the Attributes of the Item are to be displayed. The rest of this document will describe how to accomplish this. To launch a "no-code" ItemView, follow the instructions in the section titled [How to launch ItemViews](#). An example of the "no-code" ItemView for the RhinoExampleCategory is shown to the right.

Analyze Item's attributes

Before writing any code or resource files for the ItemView, begin by analyzing the information that needs to be displayed. Divide the information into two groups: information that will go in the Fields section, and information that will go in the Additional Information section. While there are no absolute rules about what kind of information goes where, here are some suggestions on how to divide the information:

- The Fields section is best suited for displaying short text strings. The Additional Information section has the ability to display larger components.
- It is suggested that information that defines the identity of the Item be in the Fields section where it will be easy to find. Secondary information, such as information about relationships between the Item and other Items, can be in the Additional Information section and interested users can take the time to locate it.
- It may be desirable to place information that is static or changes infrequently in the Fields section, and put dynamic information that changes often during the normal operation of the system in the Additional Information section.
- Don't put too many pieces of information in the Fields section. More than about 7 lines will make the ItemView hard to read. Try to split the information into smaller sets that are logically and semantically grouped. Put the most important sets of information in the Fields section and put the rest in the Additional Information section.
- In each section, order the information by importance, with the most important information at the top.
- In some cases, it makes sense to break these guidelines to group common pieces of information or to give the ItemView an appealing layout that's easy to understand.

All of the information in a RhinoExampleCategory Item probably belongs in the Fields section, but imagine that the Printer type of Item also had a list of print jobs. Since the list could be quite long, using a comma separated list would not be a practical solution. In this case, it might work to use a JList to implement a scrollable list to display all of the print jobs. This component would be displayed in the Additional Information section).

Customizing the Fields of the ItemView

There are several types of properties that control the look of the ItemView. The properties (in the order that they are described) are:

1. [The *field* properties](#) - Define names of the fields.
2. [The *basedOn* properties](#) - Tell the ItemView which Attributes correspond to particular fields.
3. [The *label* properties](#) - Provide the labels the ItemView will use for the fields.
4. [The *method* properties](#) - Specify the manner in which the ItemView will use the Item's Attributes to fill in the field.

The *field* Properties

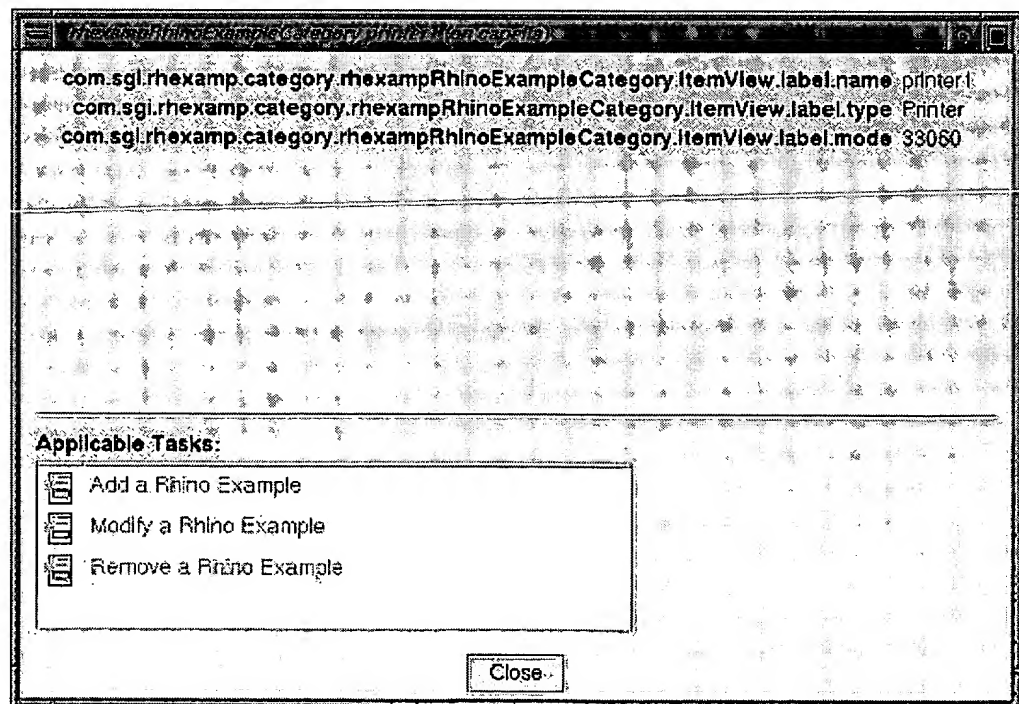
The pieces of the Item's information that are displayed in the Fields section are completely controlled by a resource file. The most fundamental resources are those that give names to the fields that will be displayed. These names identify the fields so that other resources can refer to particular fields. The resources follow the form `<Category name>.ItemView.field<n>`, where `<Category name>` is the name of the Category (see the [FIELDS](#) documentation for more info), and `<n>` represents integers starting at 0 that represent order in which the fields should be displayed. For example, the resource file that controls the `RhinoExampleCategory` contains the following lines (the letters in the first column are for reference purposes only):

```
A: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemView.field0=name
B: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemView.field1=type
C: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemView.field2=mode
```

Because the first part of each line is identical, it is common to use macros to shorten the lines of the resource file and to make the file easier to read. An example of the same resources using macros is shown below.

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: IVprefix=${RHINO_EXAMPLE_CATEGORY}.ItemView
C:
D: ${IVprefix}.field0=name
E: ${IVprefix}.field1=type
F: ${IVprefix}.field2=mode
```

The three "field" resources (D - F) define the names of the fields and the order in which the fields will be displayed in the ItemView. In this example, the names of the fields correspond exactly with the names of the Attributes in the Item that will be displayed in the field. By naming the fields in this manner, the ItemView can use default behavior and automatically



associate the correct Attribute with the field. It is also possible to give the fields names that are not the same as the names of Attributes. In that case, it may be necessary to use the "basedOn" property (defined below) to tell the ItemView which Attribute is associated with a field.

Running an ItemView with the 5 lines described above in the resource file will result in an ItemView that is shown on the right. Notice that the order of the fields is "name", "type", and then "node", which is as specified in the resource file. ItemView has used a default label for each of the fields. Information about how to customize the label is described below. The ItemView is using the default "toString" method (methods are described below). This is the simplest method, and uses the results of calling Java's toString method on the value of the Attribute.

The *basedOn* Properties

In the example resource file shown above, the names of the fields were defined to be the same as the Item's Attributes that they represented. This allowed the ItemView to automatically show the value of the Attribute in the field. It is sometimes desirable to use different names for the fields than the Attributes that they represent. This can make the resource file more readable or can be required because there may not be a one to one correspondence between the Attributes in the Item and the fields that are displayed.

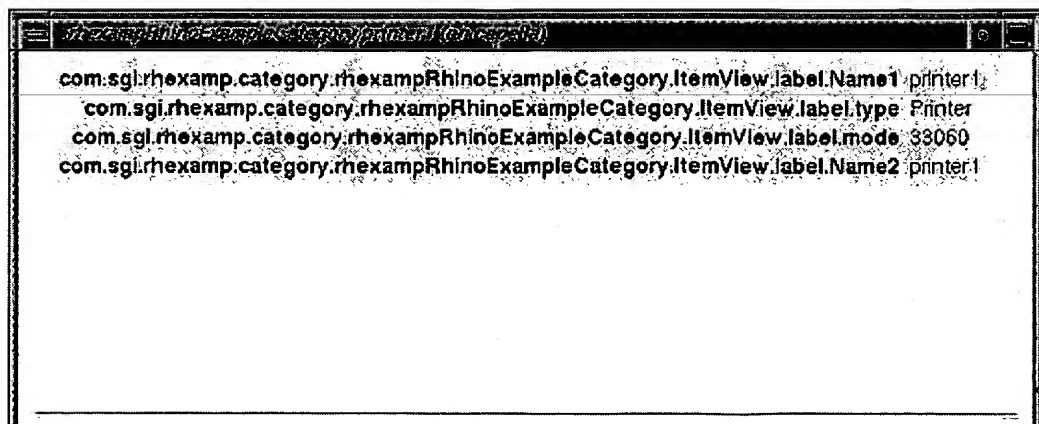
If a field is given a name that does not correspond to the name of an Attribute, the "basedOn" property is used to tell the ItemView which Attribute the field represents. The "basedOn" resources are defined as: `<Category name>.ItemView.basedOn.<field>`, where `<Category name>` is the name of the Category, and `<field>` is the name of a field. (See the BASED_ON documentation for more info).

The renderer method (as described below) does not require that the field be associated with a particular Attribute. When using this method, it is not necessary to specify the "basedOn" property even if the name of the field does not correspond to an Attribute. All the other methods, including the default "toString" method, require that the field be associated with a particular Attribute of the Item.

For example, suppose that the "name" Attribute should be displayed twice, once at the beginning of the list, and once at the end. A resource file as follows would do just that:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: IVprefix=${RHINO_EXAMPLE_CATEGORY}.ItemView
C:
D: ${IVprefix}.field0=Name1
E: ${IVprefix}.field1=type
F: ${IVprefix}.field2=node
G: ${IVprefix}.field3=Name2
H:
I: ${IVprefix}.basedOn.Name1=name
J: ${IVprefix}.basedOn.Name2=name
```

This would result in the name being shown twice, as is seen to the right:



The *label* Properties

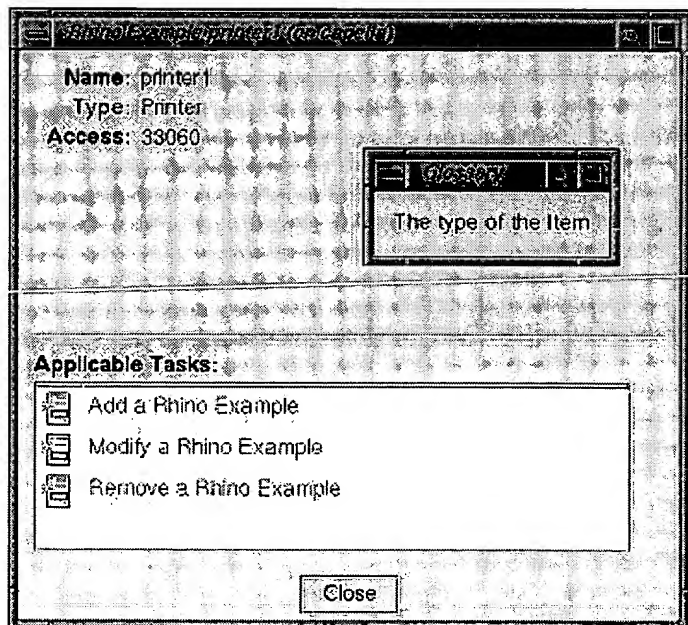
The next step is to define the strings that will be used as the labels for the fields. The "label" resources are defined as: `<Category name>.ItemView.label.<field>.label`. (See the [LABEL](#) documentation for more info). Optionally, another resource can be specified that gives the name of a glossary entry that will be displayed if the user clicks on the label. This resource is defined as:

`<Category name>.ItemView.label.<field>.glossary`, and if this resource is defined, then the label will appear blue.

For example, define labels for the example ItemView, the following properties would be added to the resource file:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: IVprefix=${RHINO_EXAMPLE_CATEGORY}.ItemView
C:
D: ${IVprefix}.field0=name
E: ${IVprefix}.field1=type
F: ${IVprefix}.field2=mode
G:
H: ${IVprefix}.label.name.label=Name:
I: ${IVprefix}.label.type.label=Type:
J: ${IVprefix}.label.type.glossary=glossary.Type
K: ${IVprefix}.label.mode.label=Access:
L:
M: glossary.Type = The type of the Item
```

Displaying the ItemView now shows that the desired labels are displayed. Notice that the "type" label is displayed as a link, and the picture shows the glossary window that results when the user clicks on the link.



The *method* Properties

The next step is to choose what method the ItemView should use to display the field. (*In this usage, "method" does not refer to a Java method, but rather to the typical English definition of the word*) The "method" resource controls this, and is defined as: `<Category name>.ItemView.method.<field>` (see the [METHOD](#) documentation for more info). Four methods are available:

1. toString

The toString method is the default method, and is what the ItemView implicitly uses to display the field if no method is specified in the properties file. The toString method calls Java's toString method on the value of the Attribute that is associated with the field (either by the "basedOn" property or the name of the field if no "basedOn" property is set). If this method is used, no additional resources are needed. For example, to make explicit the fact that the "name" field should use the toString method, include the following in the resource file:

```
${IVprefix}.method.name=toString
```

2. lookup

The lookup method uses the value of the Attribute associated with the field (either by the "basedOn" property or the name of the field if no "basedOn" property is set) as a key to lookup a string in a table of values. This is good for cases when the value of the Attributes comes from a limited set of possible values, and there is a mapping from the Attribute's value to some more easily understandable string. This method is also good when there is a need to localize the text that gets displayed in the field. If the "lookup" type is used, additional "lookup" resources (defined as *<Category name>.ItemView.lookup.<field>.<Attribute's value>*) should also be provided for each of the possible values of the Attribute. For example, to specify that the "type" field should use the lookup method, and should display the type in Spanish instead of English, include the following in the resource file:

A: `${IVprefix}.method.type=lookup`

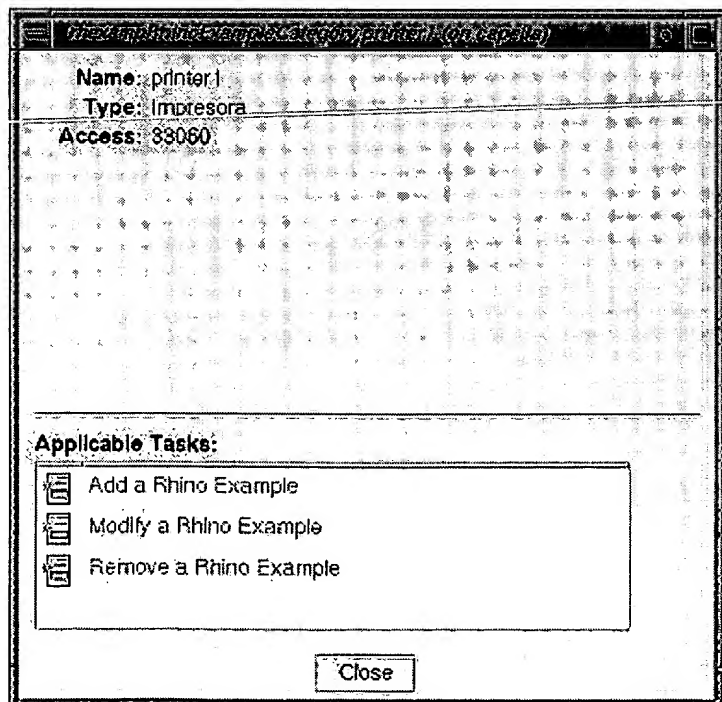
B:

C: `${IVprefix}.lookup.type.Printer=Impresora`

D: `${IVprefix}.lookup.type.Clock=Reloj`

E: `${IVprefix}.lookup.type.NetscapeExecutable=Netscape`

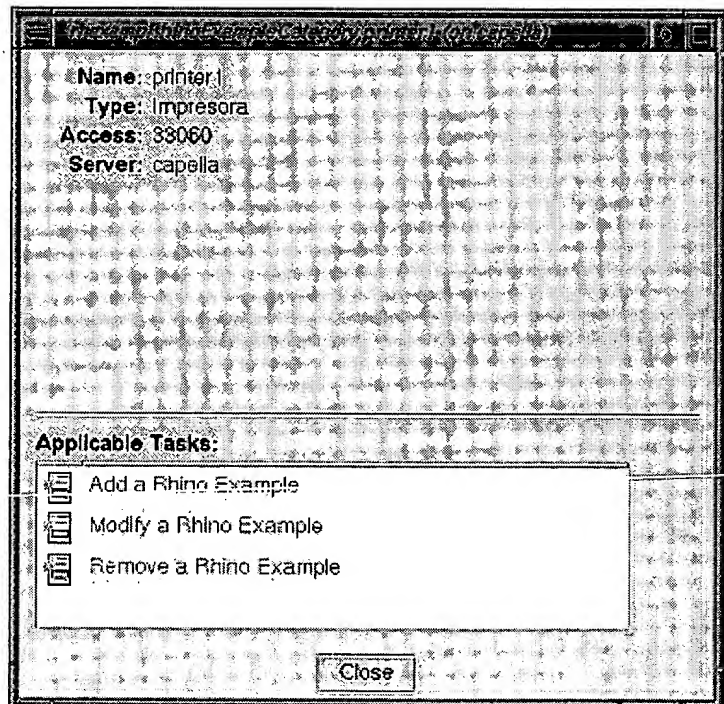
In this case, the type of the Item will be displayed in it's Spanish equivalent:



3. richText

The `richText` method will display the string value of the Attribute just as the `toString` method does, but will display it as a link that launches an ItemView. This is generally used to show the relationship between an Item in one Category and an Item in another Category. The example used in this document has one Category, but consider the case where each of the Items in the RhinoExample category had an Attribute in it that specifies the server on which the Item was running. Assume also that there is a second Category, "rhexampServerCategory" with server Items. Consider that the RhinoExample Item has an Attribute with the name "server" that is the name of the server that the RhinoExample is running on, and another Attribute "server_selector" which is the selector of the server in the "rhexampServerCategory" category. (In many cases, the name of the server would be the same as the selector of the server. In that case, substitute "server" for "server_selector" in the following example.) To show a link to the appropriate server from the RhinoExample ItemView, the following would be added to the Resource File:

```
A: ${IVprefix}.field3=server
B: ${IVprefix}.label.server=Server:
C:
D: ${IVprefix}.method.server=richText
E: ${IVprefix}.selector.server=server_selector
F: ${IVprefix}.category.server=rhexampServerCategory
```



4. renderer

It is sometimes the case that none of the three ways presented so far are adequate to display the state of the Item. Such cases can result when:

- o there is a need to synthesize two or more Attributes into a single value for display
- o Java code is needed to decode the Attribute (or Attributes) into a user-readable value
- o A special component is needed to display text
- o It is desired to show a label with color
- o The user would want launch something other than an ItemView
- o any thing else not permitted with the three predefined methods

In any of these cases, the `renderer` method should be used. This method provides a chance to write a small piece of Java code that will control the display of the field. In the case of the

RhinoExample Category, the renderer converts the numeric "mode" Attribute into text that is displayed to the user. For example, the mode "33060" is displayed as "Read Only". See the RhinoExampleCategoryRenderers file for this example.

The *missing* properties

Depending on the way that the server-side Category is written, there may be cases where a particular Attribute is missing from an Item. For example, consider that the Item can optionally contain the "type" Attribute. If the Item contains that Attribute, then the ItemView should display the name using the lookup method as described above. Otherwise, the ItemView should display some other string, such as "(Unknown)". For this situation, you can use the "missing" resource (defined as `<Category name>.ItemView.missing.<field>`). The "missing" resource allows you to specify a string that will be displayed if an Attribute is missing from an Item. The "missing" resource can be used with the `toString`, `lookup`, or `richText` methods.

For example, to use the string "(Desconocido)" (Spanish for "Unknown") if the "type" Attribute is missing from the Item, add the following resource:

```
A: ${IVprefix}.method.type=lookup
B:
C: ${IVprefix}.lookup.type.Printer=Impresora
D: ${IVprefix}.lookup.type.Clock=Reloj
E: ${IVprefix}.lookup.type.NetscapeExecutable=Netscape
F: ${IVprefix}.missing.type=(Desconocido)
```

Writing an ItemViewFieldRenderer

ItemViews use an instance of the `ItemViewFieldRenderer` interface to render fields that use the `renderer` method. There is only one `ItemViewFieldRenderer` per ItemView, so it must be able to handle all of the fields in the ItemView that are using the `renderer` method. A class should be written that implements the `ItemViewFieldRenderer` interface, and placed in the product's "category" package. (The file can actually be placed anywhere, but the "category" package is one logical place). Tell the ItemView how to find the class by naming it in the property file with the "fieldRenderer" property, which is defined as `<Category name>.ItemView.fieldRenderer` (see the [FIELD_RENDERER](#) documentation for more info). For example, the RhinoExampleCategory (whose full name is `com.sgi.rhexamp.category.rhexampRhinoExampleCategory`) has a class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryRenderers` that implements the `ItemViewFieldRenderer` interface, and so the following line is included in the Category's resource file:

```
A: ${IVprefix}.fieldRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
```

If a field uses the `renderer` method, but no `ItemViewFieldRenderer` is defined with the "fieldRenderer" property, then the ItemView will attempt to load a class with the name `<Category Name>FieldRenderer`. For example, for the RhinoExampleCategory, it would attempt to load the class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryFieldRenderer`. If the "fieldRenderer" resource is not specified and the `<Category Name>FieldRenderer` class is not found, then ItemView will throw an assertion.

The `ItemViewFieldRenderer` has five methods that must be implemented. See the documentation for `ItemViewFieldRenderer` about the specifics of each method.

- `public void initializeFieldRenderer(ItemViewContext ivc, ItemViewController controller);`
- `public Component getComponentForField(String field);`
- `public void renderFields(Item item);`
- `public void renderFieldsAgain(Item item);`
- `public void renderFieldsBlank();`

The sequence that the methods will be called in is as follows:

1. `initializeFieldRenderer`
2. `getComponentForField` (once for each field using the renderer)
3. `renderFields`
4. `renderFieldsAgain` (zero or more times)
5. `renderFieldsBlank`
6. repeat from step 3 (only if `ItemView` is used to display another `Item`)

The `initializeFieldRenderer` method is responsible for initializing the renderer. The `ItemView` calls the `getComponentForField` method once for each field that is using the `ItemViewFieldRenderer`. The `ItemView` passes in the name of the field, and the renderer passes back the component that the `ItemView` should use to display the field. When the `ItemView` obtains an `Item`, it passes the `Item` to `renderFields`. At this time, the renderer should use the `Item` to fill in the information that the components are displaying. If the `Item` changes, the `ItemView` calls `renderFieldsAgain`, passing the `Item`. The renderer should update the components to show the current state. If the `Item` is deleted, or the `ItemView` is disposed, the `ItemView` will call `renderFieldsBlank`. The renderer should remove any state information from the component, and prepare itself to be garbage collected. In the case that the `Item` reappears, or the `ItemView` is set to display another `Item`, the sequence repeats from step 3 (`renderFields`).

Writing an `ItemViewAdditionalInfoRenderer`

The additional info area of the `ItemView` is completely at the discretion of the programmer, and Java code must be written to display anything. `ItemViews` use an instance of `ItemViewAdditionalInfoRenderer` to render the additional info section. Write a class that implements the `ItemViewAdditionalInfoRenderer` interface, and place it in the product's "category" package. (The file can actually be placed anywhere, but the "category" package is one logical place). Tell the `ItemView` how to find the class by naming it in the property file with the "additionalInfoRenderer" property, which is defined as `<Category name>.ItemView.additionalInfoRenderer` (see the [ADDITIONAL_INFO_RENDERER](#) documentation for more info). For example, the `RhinoExampleCategory` (whose full name is `com.sgi.rhexamp.category.rhexampRhinoExampleCategory`) has a class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryRenderers` that implements the `ItemViewAdditionalInfoRenderer` interface, and so the following line is included in the `Category`'s resource file:

```
A: ${IVprefix}.additionalInfoRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
```

If no `ItemViewAdditionalInfoRenderer` is defined with the "additionalInfoRenderer" property, then the `ItemView` will attempt to load a class with the name `<Category Name>AdditionalInfoRenderer`. For example, for the `rhinoExampleCategory`, it would attempt to load the class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryAdditionalInfoRenderer`. If the "additionalInfoRenderer" resource is not specified and the `<Category Name>AdditionalInfoRenderer` class is not found, then `ItemView` will not display anything in the "Additional Information" section.

The API for the `ItemViewAdditionalInfoRenderer` is almost identical to that of the `ItemViewFieldRenderer`. In this case, there are four methods that must be implemented:

- `public void initializeAdditionalInfoRenderer(LabelComponentPanel panel, ItemViewContext ivc, ItemViewController controller);`
- `public void renderInfo(Item item);`
- `public void renderInfoAgain(Item item);`
- `public void renderInfoBlank();`

The sequence that the methods will be called in is as follows:

1. `initializeAdditionalInfoRenderer`
2. `renderInfo`
3. `renderInfoAgain` (zero or more times)
4. `renderInfoBlank`
5. repeat from step 2 (only if `ItemView` is used to display another `Item`)

The `initializeAdditionalInfoRenderer` method is responsible for initializing the renderer and preparing it for use. The `ItemView` passes the method a `LabelComponentPanel` that the renderer should add its components to. When the `ItemView` receives an `Item`, it passes the `renderInfo` method the `Item`, and the renderer should update the components on the panel as it wishes. If the `Item` changes its state then the `ItemView` will call `renderInfoAgain` and the renderer should update all of the components to show the current state. If the `Item` is deleted or the `ItemView` is disposed, then the `ItemView` will call `renderInfoBlank`, and in response the renderer should update the components to not show any state and prepare to be garbage collected. In the case that the `Item` reappears, or the `ItemView` is set to display another `Item`, the sequence repeats from step 2 (`renderInfo`).

Controlling the Icon displayed for an ItemView

The `ItemView` does not directly control the Icon that is displayed. The Icon is generated by the `ResourceBasedIconRenderer`. See the tutorial on using `ResourceBasedIconRenderer` for details on how to control the Icon.

Controlling the title of the ItemView (as displayed on the window's title bar)

The `ItemView` does not directly control the title that is used. The Title is generated by the `ResourceBasedNameRenderer`. See the tutorial on using `ResourceBasedNameRenderer` for details on

how to control the title.

Controlling the TaskShelf on the ItemView

The ItemView does not directly control the Tasks shown in the TaskShelf. The Tasks are generated by the TaskRegistry.

How to launch ItemViews

To view an ItemView from the command line, type:

```
%>
java com.sgi.sysadm.manager.RunItemView <package qualified Category name> <Item sele
```

For example, to launch an ItemView of the Item "foo" in Category "BarCategory", where the ItemView's resource file is in com/sgi/myProduct/category (relative to classpath), type:

```
%> java com.sgi.sysadm.manager.RunItemView com.sgi.myProduct.category.BarCategory
foo
```

To launch a "no-code" ItemView, pass the Category selector instead of the fully qualified name:

```
%> java com.sgi.sysadm.manager.RunItemView BarCategory foo
```

A "no-code" ItemView will also be displayed if no resources corresponding to the Category are found.

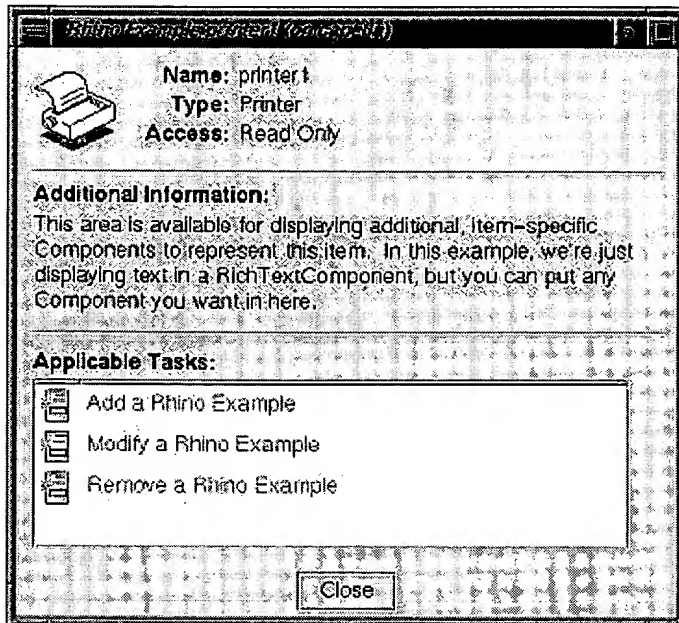
To programmatically launch an ItemView, use one of two methods: To launch an ItemView in a new frame (called an ItemViewFrame), use the launchItemViewFrame method in ItemViewFrame. The launchItemViewFrame method takes a ItemViewLaunchRequestEvent, which encapsulates all the information about which ItemView to launch. For example:

```
1:  ItemViewFrame.launchItemViewFrame(
2:      new ItemViewLaunchRequestEvent(this,
3:                                     "com.sgi.myProduct.category.BarCategory",
4:                                     "foo"),
5:      new UIContext());
```

To embed an ItemView in another component, create an ItemView with the createItemView method of ItemView, set Item to display with the setItem method, then call getPanel on ItemView to get a panel that contains the ItemView. For example:

```
1:  ItemView iv = ItemView.createItemView(_hostContext,
2:                                     "com.sgi.myProduct.category.BarCategory")
3:  iv.setItem("foo");
4:  _panel.add(iv.getPanel());
```

Typical Resource File for an ItemView



```

A:  # Set up some macros to use in this resource file.  See the
B:  # ResourceStack documentation for more about macros.
C:  RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
D:  IVprefix=${RHINO_EXAMPLE_CATEGORY}.ItemView
E:  ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
F:
G:
H:  # Set the width and height of the ItemView.  See the PANEL_WIDTH and
I:  # PANEL_HEIGHT documentation for more information.
J:  ItemViewPanel.width=333
K:  ItemViewPanel.height=260
L:
M:  # Set up the three fields.  Call the fields "name", "type", and "mode".
N:  ${IVprefix}.field0=name
O:  ${IVprefix}.field1=type
P:  ${IVprefix}.field2=mode
Q:
R:  # Tell the ItemView which Attributes of the Item to use to show the
S:  # appropriate field.  It is not necessary to set the "basedOn" resource
T:  # for a field that is using the "renderer" method, which is why there is
U:  # no "${IVprefix}.basedOn.mode" resource.  In this case, the next two
V:  # lines are unnecessary, because the name of the Attribute is the same
W:  # as the field.  They are included here to make the resource file easier
X:  # to understand and for illustration purposes.
Y:  ${IVprefix}.basedOn.name=name
Z:  ${IVprefix}.basedOn.type=type
AA:
AB:  # Sets the labels to be used for the three fields.
AC:  ${IVprefix}.label.name.label=Name:
AD:  ${IVprefix}.label.type.label=Type:
AE:  ${IVprefix}.label.mode.label=Access:
AF:
AG:  # Sets the method that the ItemView will use to display the
AH:  # three fields.

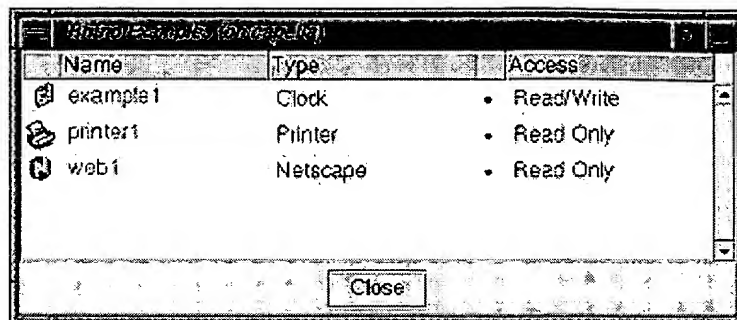
```

```
AI: ${IVprefix}.method.name=toString
AJ: ${IVprefix}.method.type=lookup
AK: ${IVprefix}.method.mode=renderer
AL:
AM: # Resources necessary because the "type" field is using the
AN: # "lookup" method. See the description of the lookup method for more informatio
AO: ${IVprefix}.lookup.type.Printer=Printer
AP: ${IVprefix}.lookup.type.Clock=Clock
AQ: ${IVprefix}.lookup.type.NetscapeExecutable=Netscape
AR:
AS: # Tells the ItemView what classes to use as the
AT: # ItemViewFieldRenderer and the ItemViewAdditionalInfoRenderer. In this
AU: # case, both renderers are in the same class, but this is not
AV: # necessarily the case.
AW: ${IVprefix}.fieldRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
AX: ${IVprefix}.additionalInfoRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
AY:
AZ: # Resources specific to the AdditionalInfoRenderer. The
BA: # AdditionalInfoRenderer has access the the ResourceStack, so this file
BB: # is a good place to put resources that control the
BC: # ItemViewAdditionalInfoRenderer or ItemViewFieldRenderer. The names of
BD: # the resources are specific to the code that is written in the
BE: # renderers.
BF: ItemView.AdditionalInfo.marginLeft=0
BG: ItemView.AdditionalInfo.marginTop=0
BH: ItemView.AdditionalInfo.marginBottom=0
BI: ItemView.AdditionalInfo.marginRight=0
BJ: ItemView.AdditionalInfo.layoutType=vertical
BK: ItemView.AdditionalInfo.label=Additional Information:
BL: ItemView.AdditionalInfo.text=This area is available for displaying \
BM: additional, item-specific Components to represent this item. In this \
BN: example, we're just displaying text in a RichTextComponent, but you can \
BO: put any Component you want in here.
```

How to write a Rhino ItemTable

Table of Contents

- [Introduction](#)
- [Overview of the ItemTable](#)
- [Before you begin](#)
- [Customizing the columns of the ItemTable](#)
 - [No-Code ItemTables](#)
 - [The *column* property](#)
 - [The *basedOn* property](#)
 - [The *label* property](#)
 - [The *method* property](#)
 - [The *toString* method](#)
 - [The *lookup* method](#)
 - [The *richText* method](#)
 - [The *icon* method](#)
 - [The *stringRenderer* method](#)
 - [The *richTextRenderer* method](#)
 - [The *componentRenderer* method](#)
 - [The *width* property](#)
 - [The *alignment* property](#)
 - [The *sort* property](#)
 - [The *missing* property](#)
- [Writing an ItemTableColumnRenderer](#)
- [Controlling the Icon displayed on the ItemTable](#)
- [Controlling the title of the ItemTable](#)
- [How to launch ItemTables](#)
- [Typical Resource File for an ItemTable](#)



Introduction

This document is a reference for SGI software engineers who will be writing ItemTables for Rhino applications. An ItemTable in the Rhino Architecture is a UI Component that displays all the Items that exists in a particular Category (or Association). The ItemTable is not meant to display all of the information about each Item (that is the job of the [ItemView](#)). The ItemTable should show the information that the user is most likely to be interested in seeing, and is limited to displaying information that can fit into the cells of the table.

Overview of the ItemTable

The ItemTable is composed of a number of columns, each column representing one piece of information from the Item. Each Item is represented by a row of the table. Each column has a descriptive header. If the user clicks on the header, the table is sorted based on the associated column.

The first column of the ItemTable is reserved for the Icon that represents the Item. The Icon is not controlled by the ItemTable, but is generated by the [ResourceBasedIconRenderer](#). See the [tutorial on using ResourceBasedIconRenderer](#) for details on how to control the Icon.

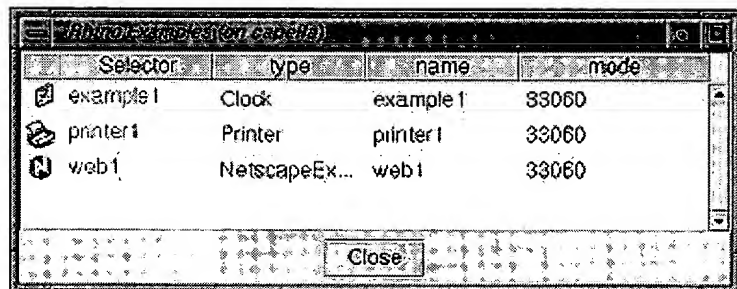
Before you begin

Before creating an ItemTable for a particular Category, it is necessary need to understand the names and terms that the Rhino infrastructure uses in relation to Categories. See the [The names of Categories on the client and on the server](#) documentation for more information.

Customizing the Columns of the ItemTable

No-Code ItemTables

While in the early stages of writing Categories, it may be desirable to show an ItemTable that shows all of the Items of a Category. The ItemTable supports this idea by means of a "no-code" ItemTable. This version of an ItemTable is not designed for use in a shipping Rhino application, but can be of great assistance while investigating the Rhino Infrastructure or for giving preliminary demos. No code or resource files need to be written to use the "no-code" ItemTable - it can be launched as soon the server side Categories have been written and the Rhino infrastructure has been installed on the client. An example of this "no-code" ItemTable is shown to the right. To launch a "no-code" ItemTable, follow the instructions in the section titled [How to launch ItemTables](#). While this ItemTable shows a lot of information, it is not suitable for use in a shipping code. To turn this ItemTable into a shippable ItemTable, it is necessary to provide resources that describe the order in which the column will be displayed, suitable headers for the columns, and ways to internationalize the ItemTable. The rest of the document will describe how to accomplish this.



Selector	type	name	mode
example1	Clock	example1	33060
printer1	Printer	printer1	33060
web1	NetscapeEx...	web1	33060

The *column* property

The manner in which the Items are displayed in the columns can be completely controlled by a resource file. The most important resource entries are the ones that name the columns that will be displayed. The names of the resources follow the form `<Category name>.ItemTable.column<n>`, where `<Category name>` is the name of the Category and `<n>` represents integers starting at 0 that represent order in which the columns should be displayed. (see the [COLUMNS](#) documentation for more details). For example, the resource file that controls the `RhinoExampleCategory` could contain the following lines (the letters in the first column are for reference purposes only):

```

A: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemTable.column0=name
B: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemTable.column1=type
C: com.sgi.rhexamp.category.rhexampRhinoExampleCategory.ItemTable.column2=mode

```

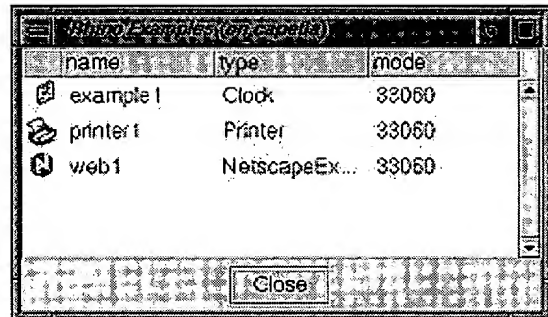
Because the first part of each line is identical, it is common to use macros to shorten the lines of the resource file and to make the file easier to read. An example of the same resources using macros is shown below.

```

A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
C:
D: ${ITprefix}.column0=name
E: ${ITprefix}.column1=type
F: ${ITprefix}.column2=mode

```

The three "column" lines (D - F) describe both the names of the columns and the order in which the columns will be displayed in the ItemTable. The names will be used later in the resource file to associate resources with particular columns. In this example, the names of the columns correspond exactly with the names of the Attributes in the Item that will be displayed in the column. By naming the columns in this manner, the ItemTable can use default behavior and automatically associate the correct Attribute with the column. It is also possible to give the columns names that are not the same as the names of Attributes. In that case, it may be necessary to use the "basedOn" property (defined below) to tell the ItemTable which Attribute is associated with a column.



Running an ItemTable with only the 5 lines described above in the resource file will result in an ItemTable that is shown on the right. Notice that the order of the columns (from left to right) is "name", "type", and then "node", which was as specified in the resource file. ItemTable has used the names of the columns as the labels for the columns. Information about how to customize the labels is described below. The ItemTable is using the default "toString" method (methods are described below). This is the simplest method, and uses the results of calling Java's toString method on the value of the Attribute.

The *basedOn* property

In the example resource file shown above, the names of the columns were defined to be the same as the Item's Attributes that they represented. This allowed the ItemTable to automatically show the value of the Attribute in the columns. It is sometimes desirable to use different names for the columns than the Attributes that control them. This can be to make the resource file more readable or because there may not be a one to one correspondence between the Attributes in the Item and the columns that are displayed.

If a column is given a name that does not correspond to the name of an Attribute, the "basedOn" property is used to tell the ItemTable which Attribute the column represents. The "basedOn" resources are defined as: `<Category name>.ItemTable.basedOn.<column>`, where `<Category name>` is the name of the Category, and `<column>` is the name of a column. (See the BASED_ON documentation for more details).

The `stringRenderer`, `richTextRenderer`, and `componentRenderer` methods (as described below) do not require that the column be associated with a particular Attribute. When using these methods, it is not necessary to specify the "basedOn" property even if the name of the column does not correspond to an Attribute. All the other methods, including the default "toString" method, require that the column be associated with a particular Attribute of the Item.

For example, suppose that for some reason we wish to display the name Attribute twice, once as the first column, and once as the last column. A resource file as follows would do just that:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
C:
D: ${ITprefix}.column0=name1
E: ${ITprefix}.column1=type
F: ${ITprefix}.column2=mode
G: ${ITprefix}.column3=name2
H:
I: ${ITprefix}.basedOn.name1=name
J: ${ITprefix}.basedOn.name2=name
```

This would result in the name being shown twice, as is seen to the right:

name1	type	mode	name2
example1	Clock	33060	example1
printer1	Printer	33060	printer1
web1	NetscapeEx...	33060	web1

The *label* property

The next step is to define the strings that will be used as the headers for the columns. The "label" resource controls this, and is defined as: `<Category name>.ItemTable.label.<column>`. (See the [LABEL](#) documentation for more details). For example, to add labels to the columns in the ItemTable, the resource file would get three new resources:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
C:
D: ${ITprefix}.column0=name
E: ${ITprefix}.column1=type
F: ${ITprefix}.column2=mode
G:
H: ${ITprefix}.label.name=Name
I: ${ITprefix}.label.type=Type
J: ${ITprefix}.label.mode=Access
```

Displaying the ItemTable now shows that the correct labels are displayed.

Name	Type	Access
example1	Clock	33060
printer1	Printer	33060
web1	NetscapeEx...	33060

The *method* property

The next step is to choose what method the ItemTable should use to display the column. (*In this usage, "method" does not refer to a Java method, but rather to the typical English definition of the word.*) The "method" resource controls this, and is defined as: `<Category name>.ItemTable.method.<column>`. (See the [METHOD](#) documentation for more details) There are seven methods available:

1. toString

The toString method is the default method, and is what the ItemTable implicitly uses to display the column if no method is specified in the properties file. The toString method calls Java's toString method on the value of the Attribute that is associated with the column (either by the "basedOn" property or the name of the column if no "basedOn" property is set). If this method is used, no additional resources are needed.

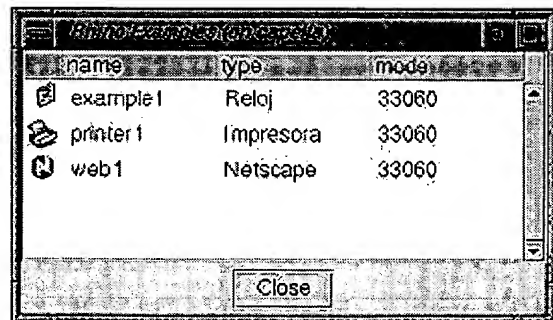
2. lookup

The lookup method uses the value of the Attribute that is associated with the column (either by the "basedOn" property or the name of the column if no "basedOn" property is set) as a key to lookup a string in a table of values. This is useful for cases when the value of the Attributes comes from a limited set of possible values, and there is a mapping from the Attribute to some more easily understandable string. This is also useful when there will be a need to localize the text that gets displayed in the column. If using the "lookup" method, also provide "lookup" resources, which are defined as `<Category name>.ItemTable.lookup.<column>.<Attribute's value>` (See the [LOOKUP](#) documentation for more details), for each of the possible values of the Attribute.

For example, to specify that the "type" column should use the lookup method, and should display the type in Spanish instead of English, include the following in the resource file:

```
A: ${ITprefix}.method.type=lookup
B:
C: ${ITprefix}.lookup.type.Printer=Impresora
D: ${ITprefix}.lookup.type.Clock=Reloj
E: ${ITprefix}.lookup.type.NetscapeExecutable=Netscape
```

In this case, the type of the Item will be displayed in it's Spanish equivalent:



name	type	mode
example1	Reloj	33060
printer1	Impresora	33060
web1	Netscape	33060

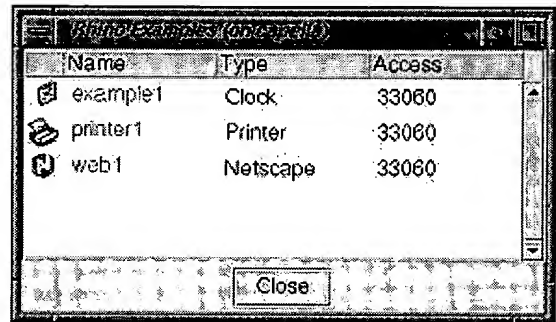
3. richText

The richText method will display the string value of the Attribute just as the toString method does, but will display it as a link that launches an ItemView. This is generally used to show the relationship between an Item in one Category and an Item in another Category, or to provide a way to launch ItemViews of the Items in the ItemTable.

When using the `richText` method for a column, there must be two additional resources defined for each column. The first is the "category" resource, which is defined as `<Category name>.ItemTable.category.<column>` (See the [CATEGORY](#) documentation for more details). The second resource is the "selector" resource, which is defined as `<Category name>.ItemTable.selector.<column>` (See the [SELECTOR](#) documentation for more details). The "category" resource is a string that gives the package-qualified name of the Category that the ItemView will use, and the "selector" resource names the Attribute whose value will be used as the selector of the Item the ItemView will show.

The example only has one Category, so to demonstrate the `richText` method, consider making the "name" column contain links to launch the appropriate ItemView. The following lines would be added the resource file:

```
A: ${ITprefix}.method.name=richText
B:
C: ${ITprefix}.category.name=${RHINO_EXAMPLE_CATEGORY}
D: ${ITprefix}.selector.name=name
```



4. icon

The `icon` method can be used to show the value of an Attribute as an icon. The value of the Attribute that is associated with the column (either by the "basedOn" property or the name of the column if no "basedOn" property is set) is turned into a string with the `toString` method, and that value is used to lookup the "icon" resource, which is defined as `<Category name>.ItemTable.icon.<column>.<Attribute's value>` (see the [ICON](#) documentation for more details). The "icon" resource should be the pathname of an icon to show in the table. If the "icon" resource is not found, then a default icon is used, which is defined as `<Category name>.ItemTable.icon.<column>` (See the [DEFAULT_ICON](#) documentation for more details). If neither the specific icon or the default icon is found, then no icon will be shown.

For example, to show an Icon that represents the "mode" Attribute, the following should be used as the resource file:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
C:
D: ${ITprefix}.column0=name
E: ${ITprefix}.column1=type
F: ${ITprefix}.column2=mode-icon
G: ${ITprefix}.column3=mode-text
H:
I: ${ITprefix}.basedOn.mode-icon=mode
J: ${ITprefix}.basedOn.mode-text=mode
K:
```

```

L: ${ITprefix}.method.mode-icon=icon
M: ${ITprefix}.icon.mode-icon.33188=/com/sgi/rhexamp/category/images/blue-ball.
N: ${ITprefix}.icon.mode-icon.33060=/com/sgi/rhexamp/category/images/red-ball.g
O: ${ITprefix}.icon.mode-icon=/com/sgi/rhexamp/category/images/yellow-ball.gif

```

The example to the right shows the ItemTable that will result. Note that because the mode needed to be displayed in two columns, once as an icon and once as text, it was necessary to add a forth column to the ItemTable and provide new names for the third and forth columns. Also notice the "basedOn" properties that tell ItemTable that the *mode* Attribute of the Item controls both columns.

name	type	mode-text
example1	Clock	33188
printer1	Printer	33060
web1	NetscapeEx...	33060

5. **stringRenderer**
6. **richTextRenderer**
7. **componentRenderer**

It is sometimes the case that none of the three ways presented so far are adequate to display the state of the Item. Such cases can result when:

- o there is a need to synthesize two or more Attributes into a single value for display
- o Java code is needed to decode the Attribute (or Attributes) into a user-readable value
- o a special component is needed to display text
- o It is desired to show a label with color
- o the user would want to launch something other than an ItemView
- o anything else not permitted with the three predefined methods

In any of these cases, use one of the renderer methods. These methods provide a chance to write a small piece of Java code that will control the display of the column. There are three types of renderers:

- o stringRenderer
- o richTextRenderer
- o componentRenderer

These renderers all use the same instance of ItemTableColumnRenderer to render the column. The renderers differ only in the type of Object that the renderer returns. In the case of the RhinoExample Category, the "mode-text" column uses a string renderer to convert the "mode" attribute of the Item into a user-readable string. See the RhinoExampleCategoryRenderers file for the example. Also see the Writing an ItemTableColumnRenderer section below about how to write an ItemTableColumnRenderer. Adding the following lines to the resource file tells the ItemTable to use the stringRenderer method for the "mode-text" column, and to use the com.sgi.rhexamp.category.rhexampRhinoExampleCategoryRenderers class as the ItemTableColumnRenderer. Lines "D" and "E" are resources that the ItemTableColumnRenderer uses.

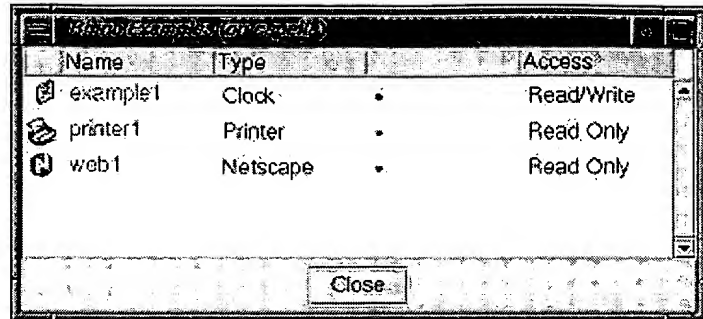
```

A: ${ITprefix}.method.mode-text=stringRenderer
B: ${ITprefix}.columnRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
C:
D: ${ITprefix}.modeStr.readWrite=Read/Write

```

E: `${ITprefix}.modeStr.readOnly=Read Only`

After adding the resources to use the string renderer, the ItemTable looks like:

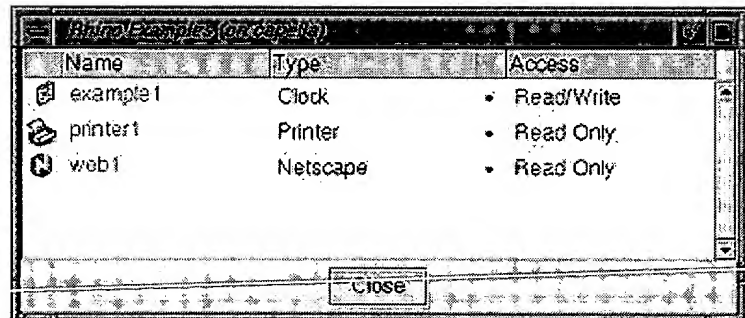


The *width* property

To set the width of a column (in points), use the "width" property, which is defined as `<Category name>.ItemTable.width.<column>`. (See the [WIDTH](#) documentation for more details). For example, to set the widths of the columns in the ItemTable, the resource file would get four new resources:

A: `${ITprefix}.width.name=100`
 B: `${ITprefix}.width.type=100`
 C: `${ITprefix}.width.mode-icon=10`
 D: `${ITprefix}.width.mode-text=100`

After setting the widths, the ItemTable is as shown:

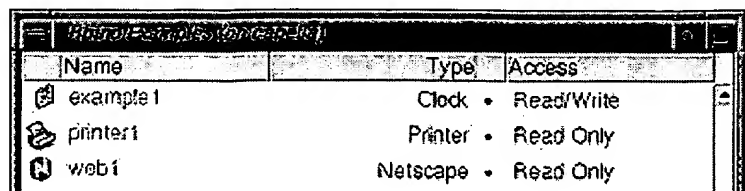


The *alignment* property

To control the alignment (justification) of the columns in the ItemTable, use the "alignment" property, which is defined as `<Category name>.ItemTable.alignment.<column>`. (See the [ALIGNMENT](#) documentation for more details). For example, to set the alignment of the columns in the ItemTable, the resource file would get two new resources:

A: `${ITprefix}.alignment.type=right`
 B: `${ITprefix}.alignment.mode-text=left`

There are no alignment resources for the name or mode-icon columns because alignment is only available on columns using the toString, lookup, and stringRenderer methods.



After setting the alignment resources as shown above, the ItemTable looks like:

The *sort* property

To control the way the the ItemTable sorts a column, use the the "sort" property, which is defined as `<Category name>.ItemTable.sort.<column>`. (See the [SORT](#) documentation for more details). There are four sorting option available, and each of them work with particular methods:

- lexical (toString, lookup, richText, stringRenderer, or richTextRenderer methods)
- numeric (toString, lookup, richText, stringRenderer, or richTextRenderer methods)
- none (all methods)
- sorter (all methods)

The "lexical" sort is a alphanumeric sort that uses the `java.text.Collator.compare` method to compare Attributes. The "numeric" sort turns the Attributes into instances of `java.lang.Integer` and then performs a numeric sort. The "none" sort specifies that there is no sort order for a column. The "sorter" sort specifies that the ItemTable should call the `compareItemsForItemTable` method of the `ItemTableColumnRenderer` to compare Items. See the [Writing an ItemTableColumnRenderer](#) section below about how to write a `ItemTableColumnRenderer`. For example, to set the sort method of the columns in the ItemTable, the resource file would get four new resources:

```
A: ${ITprefix}.sort.name=lexical
B: ${ITprefix}.sort.type=lexical
C: ${ITprefix}.sort.mode-icon=sorter
D: ${ITprefix}.sort.mode-text=lexical
```

The *missing* property

Depending on the way that the server-side Category is written, there may be cases where a particular Attribute is missing from an Item. For example, consider that the Item can optionally contain the "type" Attribute. If the Item contains that Attribute, then the column should display the name using the `lookup` method as described above. Otherwise, the column should display some other string, such as "(Unknown)". For this situation, you can use the "missing" resource (defined as `<Category name>.ItemTable.missing.<column>`). The "missing" resource allows you to specify a string that will be displayed if an Attribute is missing from an Item. The "missing" resource can be used with the `toString`, `lookup`, or `richText` methods.

For example, to use the string "(Desconocido)" (Spanish for "Unknown") if the "type" Attribute is missing from the Item, add the following resource:

```
A: ${ITprefix}.method.type=lookup
B:
C: ${ITprefix}.lookup.type.Printer=Impresora
D: ${ITprefix}.lookup.type.Clock=Reloj
E: ${ITprefix}.lookup.type.NetscapeExecutable=Netscape
F: ${ITprefix}.missing.type=(Desconocido)
```

Writing an ItemTableColumnRenderer

ItemTables use an instance of the `ItemTableColumnRenderer` interface to render columns that are using the `stringRenderer`, `richTextRenderer`, and `componentRenderer` methods. There is only one `ItemTableColumnRenderer` per `ItemTable`, so it must be able to handle all of the columns in the `ItemTable` that are using a renderer method. Write a class that implements the `ItemTableColumnRenderer` interface, and place it in the product's "category" package. (The file can actually be placed anywhere, but the "category" package is one logical place.) Tell the `ItemTable` how to find the class by naming it in the property file with the "columnRenderer" property, which is defined as `<Category name>.ItemTable.columnRenderer` (see the `COLUMN_RENDERER` documentation for more info). For example, the `RhinoExampleCategory` has a class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryRenderers` that implements the `ItemTableColumnRenderer` interface, and so the following line is included in the Category's resource file:

```
A: ${ITprefix}.columnRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
```

If a column uses a renderer method, but no `ItemTableColumnRenderer` is defined with the "columnRenderer" property, then the `ItemTable` will attempt to load a class with the name `<Category Name>ColumnRenderer`. For example, for the `rhinoExampleCategory`, it would attempt to load the class `com.sgi.rhexamp.category.rhexampRhinoExampleCategoryColumnRenderer`. If the "columnRenderer" resource is not specified and the `<Category Name>ColumnRenderer` class is not found, then `ItemTable` will throw an assertion.

The `ItemTableColumnRenderer` has four methods that must be implemented: See the documentation for `ItemTableColumnRenderer` about the specifics of each method.

- `public String getStringForCellofItemTable(Item item, String columnName, ItemTableContext context)`
- `public String getRichTextForCellofItemTable(Item item, String columnName, ItemTableContext context);`
- `public JComponent getComponentForCellofItemTable(Item item, String columnName, ItemTableContext context)`
- `public int compareItemsForItemTable(Item item1, Item item2, String columnName);`

When the `ItemTable` requires that a cell be rendered, it will call one of the `get*ForCellofItemTable` methods, depending on the type of renderer being used.

For the *stringRenderer* method, the `ItemTable` will call the `getStringForCellofItemTable` method, and the method should compute the String to display and return it.

For the *richTextRenderer* method, the `ItemTable` will call the `getRichTextForCellofItemTable` method, and the method should compute the String of HTML to display in a `RichTextComponent` and return it. To construct a URL that will launch an `ItemView`, use the `createUrlToLaunch` method of `ItemView`.

For the *componentRenderrer* method, the `ItemTable` will call the `getComponentForCellofItemTable`, and the method should return a `Component` that the `ItemTable` should display in the appropriate cell.

The `compareItemsForItemTable` method is used to sort the `ItemTable` based on a column that is using the "sorter" method of sorting. The `ItemTable` will pass two `Items` and the name of the column to the method, and the method should return an integer representing which of the `Items` should come first in the sorted list. See the [ItemTableColumnRenderer](#) documentation for more information about these methods.

Controlling the Icon displayed for an Item in the ItemTable

The `ItemTable` does not directly control the Icon that is displayed. The Icon is generated by the [ResourceBasedIconRenderer](#). See the [tutorial on using ResourceBasedIconRenderer](#) for details on how to control the Icon.

Controlling the title of the ItemTable

The `ItemTable` does not directly control the title that is used (as displayed on the window's title bar). The Title is generated by the [ResourceBasedNameRenderer](#). See the [tutorial on using ResourceBasedNameRenderer](#) for details on how to control the title.

How to launch ItemTables

To view an `ItemTable` from the command line, type:

```
%> java com.sgi.sysadm.manager.RunItemTable <Category Name>
```

For example, to launch an `ItemTable` for Category "BarCategory", where the `ItemTable`'s resource file is in `/com/sgi/myProduct/category` (relative to classpath), type:

```
%> java com.sgi.sysadm.manager.RunItemTable com.sgi.myProduct.category.BarCategory
```

To launch a no-code `ItemTable`, omit the name of the package:

```
%> java com.sgi.sysadm.manager.RunItemTable BarCategory
```

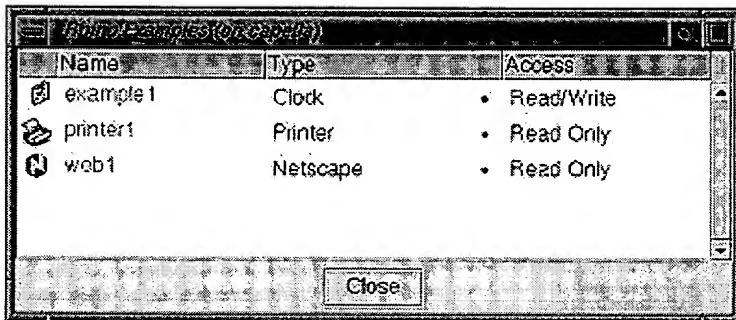
To programmatically launch an `ItemTable`, use one of two methods: To launch an `ItemTable` in a new frame (called an [ItemTableFrame](#)), use the [launchItemTableFrame](#) method in `ItemTableFrame`. The `launchItemTableFrame` method takes a [ItemTableLaunchRequestEvent](#), which encapsulates all the information about which `ItemTable` to launch. For example:

```
1: ItemTableFrame.launchItemTableFrame(  
2:     new ItemTableLaunchRequestEvent(this,  
3:                                     "com.sgi.myProduct.category.BarCategory"),  
4:     new UIContext());
```

To embed an ItemTable in another component, create an `ItemTable` with the `createItemTable` method of `ItemTable`, set `Category` to display with the `setCategory` method, then call `getItemTablePanel` on `ItemTable` to get a panel that contains the `ItemTable`. For example:

```
1:  ItemTable it = ItemTable.createItemTable(_hostContext,
2:                                          "com.sgi.myProduct.category.BarCategory");
3:  it.setCategory(_hostContext.getCategory("BarCategory"));
4:  _panel.add(it.getItemTablePanel());
```

Typical Resource File for an ItemTable



```
A:  # Set up some macros to use in this resource file.  See the
B:  # ResourceStack documentation for more about macros.
C:  RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
D:  ITprefix=${RHINO_EXAMPLE_CATEGORY}.ItemTable
E:
F:  # Define the columns to displayed.  Call them "mode", "type",
G:  # "mode-icon", and "mode-text".
H:  ${ITprefix}.column0=name
I:  ${ITprefix}.column1=type
J:  ${ITprefix}.column2=mode-icon
K:  ${ITprefix}.column3=mode-text
L:
M:  # Tells the ItemTable which Attributes of the Item to use to show the
N:  # columns.  It is not necessary to set a resource for mode-text because
O:  # it's using a "renderer" method, and basedOn is not used for the
P:  # "renderer" methods.
Q:  ${ITprefix}.basedOn.name=name
R:  ${ITprefix}.basedOn.type=type
S:  ${ITprefix}.basedOn.mode-icon=mode
T:
U:  # Sets the method that the ItemTable will use to display the four columns
V:  ${ITprefix}.method.name=richText
W:  ${ITprefix}.method.type=lookup
X:  ${ITprefix}.method.mode-icon=icon
Y:  ${ITprefix}.method.mode-text=stringRenderer
Z:
AA: # Additional resources that are necessary because the "mode-icon"
AB: # column is using the "icon" method.
AC: ${ITprefix}.icon.mode-icon.33188=/com/sgi/rhexamp/category/images/blue-ball.gif
AD: ${ITprefix}.icon.mode-icon.33060=/com/sgi/rhexamp/category/images/red-ball.gif
AE: ${ITprefix}.icon.mode-icon=/com/sgi/rhexamp/category/images/yellow-ball.gif
```

```
AF:
AG: # Additional resources that are necessary because the "type" column
AH: # is using the "lookup" method.
AI: ${ITprefix}.lookup.type.Printer=Printer
AJ: ${ITprefix}.lookup.type.Clock=Clock
AK: ${ITprefix}.lookup.type.NetscapeExecutable=Netscape
AL:
AM: # Additional resources that are necessary because the "name" column
AN: # is using a "richText" method.
AO: ${ITprefix}.category.name=${RHINO_EXAMPLE_CATEGORY}
AP: ${ITprefix}.selector.name=name
AQ:
AR: # Sets the labels that will be used for the columns.
AS: ${ITprefix}.label.name=Name
AT: ${ITprefix}.label.type=Type
AU: ${ITprefix}.label.mode-icon=
AV: ${ITprefix}.label.mode-text=Access
AW:
AX: # Sets the widths of the columns
AY: ${ITprefix}.width.name=100
AZ: ${ITprefix}.width.type=100
BA: ${ITprefix}.width.mode-icon=10
BB: ${ITprefix}.width.mode-text=100
BC:
BD: # Sets the alignment that will be used for the columns. There are
BE: # no alignment resources for the name or mode-icon columns because
BF: # alignment is only available on columns using the toString, lookup, and
BG: # stringRenderer methods.
BH: ${ITprefix}.alignment.type=left
BI: ${ITprefix}.alignment.mode-text=left
BJ:
BK: # Sets the type of sort that will be used for the columns.
BL: ${ITprefix}.sort.name=lexical
BM: ${ITprefix}.sort.type=lexical
BN: ${ITprefix}.sort.mode-icon=sorter
BO: ${ITprefix}.sort.mode-text=lexical
BP:
BQ: # Tells the ItemTable what class to use as the ItemTableColumnRenderer.
BR: ${ITprefix}.columnRenderer=${RHINO_EXAMPLE_CATEGORY}Renderers
BS:
BT: # Resources used by the ItemTableColumnRenderer.
BU: ${ITprefix}.modeStr.readWrite=Read/Write
BV: ${ITprefix}.modeStr.readOnly=Read Only
```

The Rhino TreeViewPane Component

Introduction

The Rhino TreeViewPane is a Component which displays a set of hierarchical data in an outline form. Multiple trees can be displayed, one at a time. The individual nodes in each tree are Items; each level in each tree contains Items within a Category or Association. The structure of the trees are specified almost entirely in Properties Files.

About the TreeViewPane

The TreeViewPane extends the JScrollPane class, and can thus be displayed within any Frame. The tree in the TreeViewPane is a JTree. Each node in the tree is associated with an Item in a particular Category or Association. Each node in the tree has an Icon and a name. The Icon can be a FtrIcon and thus can visually respond to changes in the state of its associated Item. It is possible to customize the display of the name of the Item by specifying an ItemNameRendererFormat.

The JTree in the TreeViewPane is also accessible so that one can take advantage of all of its capabilities (including listening for selections).

Creating a TreeViewPane

As with most Rhino UI classes, there are two basic steps in the creation of a TreeViewPane. First, one adds properties to the properties file which define the structure of the tree. Second, one writes the code which creates a new TreeViewPane which is defined by those properties. The correlation between the properties and the TreeViewPane is a name, a String, which is used as a prefix to the various property names. Pass this string to the TreeViewPane constructor as the *prefix* argument.

The TreeViewPane Properties

The various properties which define the structure of the tree are shown below. Default values, if any, are shown in parentheses.

General Appearance:

`<prefix>.background`

Specifies the color to be used as the background for the TreeViewPane (#99cccc).

`<prefix>.width`

Specifies the default width, in points, of the tree pane (160).

`<prefix>.height`

Specifies the default height, in points, of the tree pane (200).

`<prefix>.toolTipText`

Specifies the string to be displayed as the ToolTip text for each node in the tree.

`<prefix>.textColor`

Specifies the color to be used to display the name of the Item at each node of the tree (#0033cc).

`<prefix>.selectColor`

Specifies the color to be used as the background of the selected Item in the tree (ffffff66)

`<prefix>.rootFont`

Specifies the name of the font to be used to display the name of the item at the root node of the tree (SansSerif-bold-12).

`<prefix>.childFont`

Specifies the name of the font to be used to display the name of all items in the tree (except for the item at the root of the tree) (SansSerif-12).

`<prefix>.cellBorderWidth`

Specifies the height, in points, of the border around each item in the tree (2).

`<prefix>.cellBorderHeight`

Specifies the width, in points, of the border around each item in the tree (2).

`<prefix>.iconWidth`

Specifies the width, in points, of the icon to be displayed at each node of the tree (17).

`<prefix>.iconHeight`

Specifies the height, in points, of the icon to be displayed at each node of the tree (17).

`<prefix>.iconBlinkOnTime`

Specifies the time, in milliseconds, that a blinking icon will be visible before it blinks off again (750).

`<prefix>.iconBlinkOffTime`

Specifies the time, in milliseconds, that a blinking icon will not be visible before it blinks on again (750).

`<prefix>.openedIcon`

Specifies the package-qualified name of the icon to display when a node in the tree has children and those children are visible, that is, when the node is open (com.sgi.sysadm.ftr.OpenArrow).

`<prefix>.closedIcon`

Specifies the package-qualified name of the icon to display when a node in the tree has children and those children are not visible, that is, when the node is closed (com.sgi.sysadm.ftr.CloseArrow).

Tree Structure:

`<prefix>.tree<n>`

A string array that specifies the names of the trees to be displayed in the TreeViewPane. One tree can be displayed at a time.

`<prefix>.<treename>.level<n>`

The package-qualified name of the Category of Item at each level of the named tree. By default each level of the tree is actually an Association between the Item at the root of the particular subtree and the Category of its children. The Category at the first level of each tree must be the same, and must match the Category of the Item passed to the TreeViewPane constructor.

`<prefix>.<treename>.level<n>.useAssoc`

Specify whether or not to use an Association as the Category for the children of this level (true).

`<prefix>.<treename>.level<n>.rootFilterAttr`

If ".useAssoc" is false, specify an Attribute of the rootItem. If the value of that Attribute of the root Item of the tree matches the value of that Attribute in each item in the Category, then the item is added to the tree.

Item Rendering:

`<prefix>.<categoryName>.displayAttr`

Specify this to override the default rendering of the name of each Item in this Category. There are two ways to override the default rendering:

1. Specify an Attribute name; the value of the Attribute will be displayed as the name of the Item (the node); and
2. Specify a format string (see `java.text.MessageFormat`). The arguments are specified as a `.displayAttrArg` string array, as below.

```
<prefix>.<categoryName>.displayAttrArg<n>
```

Each `.displayAttrArg` (numbered from 0) is an Attribute name. The values of the Item Attributes are passed as arguments to `java.text.MessageFormat`.

For example, suppose the Properties file contains the following entries:

```
<prefix>.com.shoon.MyCategory.displayAttr = {0}: {1}
<prefix>.com.shoon.MyCategory.displayAttrArg0 = ITEM_TYPE
<prefix>.com.shoon.MyCategory.displayAttrArg1 = ITEM_NAME
```

And let's suppose the Item corresponding to a given node of the tree has the following Attributes:

```
ITEM_TYPE = Personal Name
ITEM_NAME = Howard
```

Then the following call will be made to render the name of the node (using the Attributes of its Item `item`):

```
java.text.MessageFormat("{0}: {1}",
                        new Object {
                            item.getValueString("ITEM_TYPE"),
                            item.getValueString("ITEM_NAME")
                        });
```

Thus the name of the Item (and the node in the tree) will be rendered as:

```
Personal Name: Howard
```

```
<prefix>.<categoryName>.stateAttr
```

The name of the Attribute of the Item to use to determine the state of the Item. The values of this Item Attribute are used to change the rendering of the icon.

```
<prefix>.<categoryName>.<state>.blink
```

Set to "true" if the icon should blink when the value of the `.stateAttr` Attribute of the Item matches "state".

```
<prefix>.<categoryName>.itemComparator
```

The fully-qualified name of a Class which is used to compare two Items in this Category. The Class must implement the ItemComparator interface.

Code to Implement a TreeViewPane

To create a new `TreeViewPane`, you must specify the Item which will serve as the root of the tree and a name which will be used to find the Properties. Note that the type of the Item must match the type of Category specified in the Properties for level0 of the tree. Here is a simple example which creates a `TreeViewPane` and adds it to the Frame (The `<prefix>` is "MyTree"):

```
TreeViewPane treeViewPane =
    new TreeViewPane(uic, hc, rs, rootItem, "MyTree");
```

```
add(treeViewPane);
```

See the description of [TreeViewPane](#) for a full description of the Class and its constructor arguments.

The tree displayed by default is tree 0 (see [Tree Structure](#) above).

To change trees programmatically, for example, using a menu, call [TreeViewPane.setTree\(int\)](#) or [TreeViewPane.setTree\(java.lang.String\)](#).

To listen for user selection of a node in the tree, use the standard JTree calls. For example, use [TreeViewPane.addTreeSelectionListener\(TreeSelectionListener\)](#) to add a listener which fires when a node in the tree is selected. You can also use [TreeViewPane.addActionListener\(ActionListener\)](#) to listen for the user performing an action upon a node in the tree.

Examples:

Here is a portion of the Properties file which defines the structure of four tree (example adapted from the FailSafe Manager 2.0 product):

```
MyTree.tree0 = groupsResources
MyTree.tree1 = resources
MyTree.tree2 = groups
MyTree.tree3 = policies

MyTree.groupsResources.level0 = com.sgi.fsmgr.category.ClusterCategory
MyTree.groupsResources.level1 = com.sgi.fsmgr.category.ResourceGroupCategory
MyTree.groupsResources.level2 = com.sgi.fsmgr.category.ResourceCategory

MyTree.resources.level0 = com.sgi.fsmgr.category.ClusterCategory
MyTree.resources.level1 = com.sgi.fsmgr.category.ResourceCategory

MyTree.groups.level0 = com.sgi.fsmgr.category.ClusterCategory
MyTree.groups.level1 = com.sgi.fsmgr.category.ResourceGroupCategory

MyTree.policies.level0 = com.sgi.fsmgr.category.ClusterCategory
MyTree.policies.level1 = com.sgi.fsmgr.category.FailoverPolicyCategory
MyTree.policies.level1.useAssoc = false

MyTree.com.sgi.fsmgr.category.ResourceCategory.displayAttr = {0}: {1}
MyTree.com.sgi.fsmgr.category.ResourceCategory.displayAttrArg0 = _RESOURCE_TYP
MyTree.com.sgi.fsmgr.category.ResourceCategory.displayAttrArg1 = _RESOURCE
MyTree.com.sgi.fsmgr.category.ResourceCategory.stateAttr = CAM_STATUS
MyTree.com.sgi.fsmgr.category.ResourceCategory.ONLINE_PENDING.blink = true
MyTree.com.sgi.fsmgr.category.ResourceCategory.OFFLINE_PENDING.blink = true
MyTree.com.sgi.fsmgr.category.ResourceCategory.ERROR.blink = true
MyTree.com.sgi.fsmgr.category.ResourceCategory.itemCompare = \
    com.sgi.fsmgr.detailView.CategoryItemCompare

MyTree.com.sgi.fsmgr.category.ResourceGroupCategory.stateAttr = CAM_STATUS
MyTree.com.sgi.fsmgr.category.ResourceGroupCategory.ERROR.blink = true
MyTree.com.sgi.fsmgr.category.ResourceGroupCategory.ONLINE_PENDING.blink = true
MyTree.com.sgi.fsmgr.category.ResourceGroupCategory.OFFLINE_PENDING.blink = true
MyTree.com.sgi.fsmgr.category.ResourceGroupCategory.itemCompare = \
    com.sgi.fsmgr.detailView.CategoryItemCompare

MyTree.com.sgi.fsmgr.category.ClusterCategory.stateAttr = CAM_STATUS
```

```
MyTree.com.sgi.fsmgr.category.ClusterCategory.INACTIVE.blink = true
```

These example Properties define four (4) trees, one of which is displayed in the TreeViewPane at any given time. The Item at the root of the tree must be in the "Cluster" Category. Any given tree can be dynamically chosen for display. To select the "resources" tree, for example, to be displayed in the TreeViewPane, the following calls are equivalent:

```
treeViewPane.setTree(1);
treeViewPane.setTree("resources");
```

The four trees which can be displayed are as follows:

1. groupsResources
This tree is three (3) levels deep. The second level of the tree is populated with Items in an Association between the root Cluster Item and Items in the "ResourceGroup" Category. The third level of the tree is populated with Items in an Association between each ResourceGroup Item at the second level of the tree and Items in the "Resource" Category.
2. resources
This tree is two (2) levels deep. The second level of the tree is populated with Items in an Association between the root Cluster Item and Items in the "Resource" Category.
3. groups
This tree is two (2) levels deep. The second level of the tree is populated with Items in an Association between the root Cluster Item and Items in the "ResourceGroup" Category.
4. policies
This tree is two (2) levels deep. The second level of the tree is populated with Items in the "FailoverPolicy" Category (no Association is used).

Four (4) Categories of Items can be displayed in the tree, as follows:

1. com.sgi.fsmgr.category.ClusterCategory
If the "CAM_STATUS" Attribute of any Item in this Category has the value "INACTIVE", its icon will blink. Names of Items in this Category will be rendered using the default rendering.
2. com.sgi.fsmgr.category.ResourceGroupCategory
If the "CAM_STATUS" Attribute of any Item in this Category has the value "ONLINE_PENDING", "ERROR", or "OFFLINE_PENDING", its icon will blink. Items in this Category will be compared (for sorting purposes) by using the ItemComparator
com.sgi.fsmgr.detailView.CategoryItemCompare. Names of Items in this Category will be rendered using the default rendering.
3. com.sgi.fsmgr.category.ResourceCategory
If the "CAM_STATUS" Attribute of any Item in this Category has the value "ONLINE_PENDING", "ERROR", or "OFFLINE_PENDING", its icon will blink. The name of the Item will be rendered using the java.text.MessageFormat string "{0}: {1}" with the arguments being the "_RESOURCE_TYPE" and "_RESOURCE" Attributes of that Item, respectively. Items in this Category will be compared (for sorting purposes) by using the ItemComparator
com.sgi.fsmgr.detailView.CategoryItemCompare.
4. com.sgi.fsmgr.category.FailoverPolicyCategory
Names of Items in this Category will be rendered using the default rendering.

To listen for a node in the tree being acted upon (double-clicked) by the user, the following code is used:

```

treeViewPane.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent event) {

        // get the node of the tree that has been selected
        //
        DefaultMutableTreeNode node = (DefaultMutableTreeNode)
            (((TreePath)event.getSource()).getLastPathComponent());

        // ... node actions go here ...

        try {

            // get the Item that belongs to the selected node
            //
            ItemUserObject nodeInfo
                = (ItemUserObject)node.getUserObject();
            Item item = nodeInfo.getItem();

            // ... actions upon the Item go here ...

        } catch (ClassCastException ex) {
        }
    }
});

```

To listen for a node in the tree being selected by the user, the following code is used:

```

treeViewPane.addTreeSelectionListener(new TreeSelectionListener() {
    public void valueChanged(TreeSelectionEvent event) {

        // get the node of the tree that has been selected
        //
        DefaultMutableTreeNode node = (DefaultMutableTreeNode)
            (event.getPath().getLastPathComponent());

        // ... node actions go here ...

        try {

            // get the Item that belongs to the selected node
            //
            ItemUserObject nodeInfo
                = (ItemUserObject)node.getUserObject();
            Item item = nodeInfo.getItem();

            // ... actions upon the Item go here ...

        } catch (ClassCastException ex) {
        }
    }
});

```

\$Revision: 1.2 \$ \$Date: 2000/09/10 08:16:57 \$

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

How to Customize the Task Manager

The **Task Manager** is a front end to all of the Tasks in a Rhino-based product. The Task Manager is customized for each Rhino product, but the basic interface appears the same for each product so that Users are presented with a common look and feel. Basic customization is accomplished through the creation of a product-specific properties file. It is also possible to plugin Java classes to handle more complex situations.

The Task Manager window has four parts, each of which can be customized to some degree.

- **Frame Title**

By default, the Task Manager Frame Title will display a static, customizable string that includes the name of the server. The Frame Title can also be customized to display arbitrary dynamic strings such as server state information.

- **Table of Contents**

On the left-hand side of the Task Manager window is the *Table of Contents* panel. The Table of Contents displays a set of links to product-specific pages. For example, a typical Table of Contents contains links to an Overview page, Search page, and a set of pages containing logically related Tasks.

- **Display Area**

When the User chooses one of the page links in the Table of Contents, the corresponding page is displayed on the right-hand side of the Task Manager window called the *Display Area*. The Display Area can display three types of pages:

1. **Text Page**

A *Text Page* contains informational text. For example, the Overview page is typically a text description of how the Task Manager for the product works and describes the other pages available.

2. **Task List Page**

A *Task List Page* contains links to logically-related Tasks and Tasksets. When any of these links is activated, the corresponding Task or Taskset is launched in a separate window.

3. **Class Page**

A *Class Page* is a page that is implemented as a Java class and plugged in by the developer of a specific product. For example, Rhino provides a Search page plugin.

- **Button Bar**

At the bottom of the Task Manager window is a button bar. A *Close* button is provided by default and will always appear as the rightmost button. Product-specific buttons may be added to the button bar.

For the duration of this document, let us assume that you are customizing the Task Manager for the product: *Rhino Example*.

The Task Manager Properties File

Basic customization of Task Manager is accomplished through the product-specific properties file called

TaskManagerP.properties. This file typically resides in the top of your package hierarchy. For example,

```
myWorkArea/package/com/sgi/rhexamp/TaskManagerP.properties
```

The property names are documented in the Rhino class [TaskManagerProperties](#) and default values, when provided, exist in `com.sgi.sysadm.manager.TaskManagerP.properties`.

Customizing the Table of Contents

First you will want to customize the header, or title shown in the Table of Contents panel. This is accomplished by defining a property in `TaskManagerP.properties` as follows:

```
TaskManager.TOC.title = <B>Rhino Example Manager</B>
```

Next, you will want to create the product-specific list of page links. This is accomplished via an ordered property set called `TaskManager.TOC.item<n>`, where each item represents one page link or a separator. For each page, you specify a page type, a page title, and a page target.

For example, to specify a text overview page as the first page, you would add the following properties to the `TaskManagerP.properties` file:

```
TaskManager.TOC.item0 = text
TaskManager.TOC.item0.title = Overview
TaskManager.TOC.item0.target = \
    <P> Rhino Example Manager Graphical User Interface \
    provides access to the tasks that help you set up and \
    administer your Rhino Example objects. \
    <P>The tasks are organized into the categories \
    described below. To view a category, click on it in the \
    <P> RhinoExample Task Manager Graphical User Interface (GUI) \
    provides access to the tasks that demonstrate use of \
    the Rhino infrastructure. \
    <P> \
    The tasks are organized into the categories described below. \
    To view a category, click on it in the column at left.<P>\
    <B>Overview</B> -- \
    Display this overview document. <P> \
    <B>Search</B> -- \
    Use keywords to search for a specific task. <P>\
    <B>RhinoExample Tasks</B> -- \
    Example tasks that demonstrate the use of the Rhino infrastructure.
```

Note that for text pages, the 'target' contains the actual text to be displayed on the page.

Now let's imagine that you want the next page to be a search page that allows Users to find the Task they are interested in via keyword. Rhino provides a Class Page plugin [SearchPanel](#) that indexes all of the Tasks and Tasksets by keyword. Here's how the page would be specified in `TaskManagerP.properties`:

```
TaskManager.TOC.item1 = class
TaskManager.TOC.item1.title = Search
TaskManager.TOC.item1.target = \
    com.sgi.sysadm.manager.taskManager.SearchPanel
```

The target for a page of type *class* is the CLASSPATH relative name of the page plugin, which must implement the TaskManagerPanel interface.

Next we'll add a separator in the Table of Contents, which does not require the title or target specifiers:

```
TaskManager.TOC.item2 = separator
```

Finally, we'll add a Task List page. Task List pages display an optional list of Tasksets (also known as Metatasks), a separator, and then an optional list of Task Groups. Tasksets provide guidance in accomplishing a high level goal that may involve multiple tasks. Task Groups are (possibly ordered) lists of Tasks that are closely related, usually by the type of object they operate on. For example, Tasks that all operate on User accounts are likely to be in the same Task Group.

```
TaskManager.TOC.item3 = tasklist
TaskManager.TOC.item3.title = RhinoExample Tasks
TaskManager.TOC.item3.target = RhinoExampleTasks
```

The target for a Task List page will be used as a key to optional property sets that describe the list of Tasksets (also known as Metatasks) and Task groups to display on the page. For example, the RhinoExample Tasklist page will have a single Task group:

```
TaskManager.RhinoExampleTasks.taskGroup0 = MyTaskGroup
TaskManager.RhinoExampleTaskGroup.introText = \
    <B>Rhino Example Tasks</B>
```

The introText property is the text to display at the top of the Task Group. The Task Group itself is installed in the TaskRegistry on the server, in a directory named "MyTaskGroup". See Plugging in a Task Group later in this document for details.

If we also wanted to display a list of Tasksets on the RhinoExample Tasks page, the properties might look like this:

```
TaskManager.RhinoExampleTasks.metatasksText = The following tasksets can \
    help you keep your system up and running in production \
    mode. Find the taskset that suits \
    your needs, then click to launch it.

TaskManager.RhinoExampleTasks.metataskItem0 = \
    com.sgi.rhexamp.metatask.FirstExampleTaskset
TaskManager.RhinoExampleTasks.metataskItem1 = \
    com.sgi.rhexamp.metatask.SecondExampleTaskset
```

The metatasksText is displayed above the entire list of Taskset links (and is optional).

Each metataskItem refers to another Properties file that describes the contents of the Taskset, which will launch in a separate window when activated. Here is an example of the contents of a Taskset properties file:

```
#
# Properties for First Example Taskset
#
Metatask.name = First Example Taskset
Metatask.keywords = keywords to help users find this Taskset
```

```

Metatask.text = \
    <B>Achieve a High Level Goal</B><P> \
    This taskset lists different ways to achieve a goal. \
    Here are some of the options: \
    <UL>\
    <LI><A href=task.com.sgi.fsmgr.task.ModifyClusterTask>
    <B>Modify an Example</B></A> -- \
    Set Example attributes \
    <LI><A href=task.com.sgi.fsmgr.task.DefineMachineTask>
    <B>Define an Example</B></A> \
    -- Create an Example. \
    </UL>

```

A Taskset (or Metatask) has three attributes set in its properties file: the name, keywords, and text. The text specified will be used to create a RichTextComponent that can contain links that launch Tasks, other Tasksets, or glossary entries when activated.

The font and color of the Table of Contents panel, title label, and page links can all be customized as well. See TaskManagerProperties for details.

Plugging in a Task Group

A Task List page can contain one or more Task Groups. Each Task Group is a (possibly ordered) list of Tasks that are closely related, usually by the type of object they operate on. For example, Tasks that all operate on User accounts are likely to be in the same Task Group.

Task Groups are specified in the properties for a Task List page rather than the Tasks themselves to allow new Tasks to be plugged in to the Task Manager without requiring the images to be re-shipped. For example, a new product could be created that adds Tasks to existing Task Groups, and they will automatically appear in the Task Manager the next time it is launched.

Tasks are plugged into Task Groups via the Task Registry on the server. Here are the steps needed to create a new Task Group named myTaskGroup in the Task Registry and add an ordered set of Tasks to that group:

1. cd myWorkarea/taskRegistry
2. mkdir MyTaskGroup
3. copy the Makefile from any existing Task Group or Task Category into MyTaskGroup (you should have an existing Task Group and Task Category if you created your Rhino ism using the Make Rhino Ism Task).
4. cd MyTaskGroup
5. For each Task you want to add:
 1. touch {4-digit order key}.com.sgi.{myProduct}.task.{taskName}
 2. p_modify -f {new file name from previous step}
6. p_modify Makefile
7. Edit the Makefile, replacing the existing file names with the new files added in the previous steps.
8. cd ..
9. p_modify Makefile
10. add MyTaskGroup to the list of directories to build
11. cd myWorkarea/build
12. update the idb with the new Tasks

13. p_integrate
14. p_finalize

For example, if you wanted to add three Tasks to the MyTaskGroup named "AddRhinoExampleTask", "ModifyRhinoExampleTask" and "DeleteRhinoExampleTask", in that order, you would do the following:

- cd myWorkarea/taskRegistry
- mkdir myTaskGroup
- cp {otherTaskGroup}/Makefile MyTaskGroup
- cd MyTaskGroup
- touch 1000.com.sgi.myProduct.task.AddRhinoExampleTask
- touch 2000.com.sgi.myProduct.task.ModifyRhinoExampleTask
- touch 3000.com.sgi.myProduct.task.DeleteRhinoExampleTask
- update the Makefiles and the idb file

You will need to build and install the taskRegistry onto your server.

Adding Buttons to the Button Bar

By default the Button Bar at the bottom of the TaskManager window has a Close button. When pressed, the Close button will terminate the Task Manager application but any other windows launched from Task Manager will stay open. The Close button will always appear as the rightmost button.

Buttons are added by creating an ordered property set in the Task Manager properties file. For example:

```
TaskManager.buttonItem0 = First Button
TaskManager.buttonItem0.target = com.sgi.myProduct.myFirstPlugin
TaskManager.buttonItem1 = Second Button
TaskManager.buttonItem1.target = com.sgi.myProduct.mySecondPlugin
```

Each button is given a name that will be displayed on that button as well as a target class that should be launched when the button is activated. The target class must implement one of TaskManagerFrame or TaskManagerAction. The first button added will be the leftmost button on the button bar. Subsequent buttons will be added to the right of the previous button but always to the left of the Close button.

Customizing the Task Manager Frame Title

By default, the Task Manager Frame Title will display a static string that includes the server name. This static string may be customized via a property. For example:

```
TaskManager.frameTitle = RhinoExample Manager (on {0})
```

Where the argument {0} is replaced with the server name.

If you wish to have a dynamic title that, for example, changes when the state of an object on the server changes, then you will want to use a TaskManagerTitleRenderer. A title renderer is a class that is responsible for keeping the title string up to date. It can monitor the server and make updates as desired. To plugin a title renderer, you use a property in the Task Manager properties file. For example:

```
TaskManager.titleRenderer = com.sgi.myProduct.plugin.MyTitleRenderer
```

Adding Code that Runs at Startup

Some products need to run initialization code when their Task Manager starts up. For example, a product might want to set up default values for TaskData that will be used by product-specific tasks. A TaskManagerInitPlugin is where that default-setting code should reside. Multiple TaskManagerInitPlugins can be plugged in via the Task Manager properties file. For example:

```
TaskManager.initPlugins0 = com.sgi.myProduct.plugin.MyFirstInitPlugin
TaskManager.initPlugins1 = com.sgi.myProduct.plugin.MySecondInitPlugin
```

The initialization plugins will be run, in order, as the first operation when the Task Manager is launched.

Running Task Manager

Let's suppose you have created your TaskManagerP.properties file in myWorkArea/package/com/sgi/myProduct and CLASSPATH includes "myWorkArea/package". To launch your customized Task Manager, you would enter the following command:

```
java com.sgi.sysadm.manager.TaskManager -p com.sgi.myProduct
```

To launch the Task Manager programmatically, you need to know the CLASSPATH relative name of the product (so that Task Manager can find the product-specific properties file. For example:

```
void launchTaskManager() {
    // Go busy while launching
    _uic.busy(new ResultListener() {
        public void succeeded(ResultEvent event) {
            TaskManager tMgr = new TaskManager("com.sgi.myProduct");
            tMgr.initApp();
            tMgr.run(_hc, new RApp.RAppLaunchListener() {
                public void launchSucceeded(RApp.RAppLaunchEvent event) {
                    _uic.notBusy();
                }
                public void launchFailed(RApp.RAppLaunchEvent event) {
                    _uic.notBusy();
                }
                public void launchAlreadyRunning(
                    RApp.RAppLaunchEvent event) {
                    _uic.notBusy();
                }
            });
        }
        public void failed(ResultEvent event) {
        }
    });
}
```

See RApp for more information on launching a Rhino application.



ope:

▼ Rhino

[Overview](#)[News](#)[Documentation](#)[FAQ](#)[Mailing List](#)[How To Contribute](#)[To-Do List](#)[Download](#)[License](#)

▼ SGI Open Source

[Project List](#)

▼ SGI Support and Resources

[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)

▼ Community Resources

[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)

Rhino

URL: <http://oss.sgi.com/projects/rhino/>

Rhino is an infrastructure for building applications that configure, manage, and monitor hardware software. Rhino provides a common, consistent, task-based, internationalized graphical user interface (GUI), with built-in command-line interfaces (CLIs) that system administrators can use to write scripts. Rhino applications consist of two parts:

- **Client-side GUI in Java.** The GUI runs on any platform that has a Java virtual machine, ; doesn't run as root or do setuid root. It can enable the user to perform a single task; it can provide an organized collection of tasks (with a built-in search mechanism); and it can include GUIs for monitoring the system.
- **Server-side daemon and command-line interfaces.** These can be written in C++ so Java doesn't have to run on the server being administered.

See the [FAQ](#) for more detailed information on Rhino, and [News](#) for information on what kind of work being done with Rhino. It's pretty big, and can benefit from a wide range of skills.

[terms of use](#) | [privacy policy](#) | [questions/](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark in](#)

Package Index

Other Packages

- package [com.sgi.sysadm.category](#)
- package [com.sgi.sysadm.manager](#)
- package [com.sgi.sysadm.manager.RApp](#)
- package [com.sgi.sysadm.manager.taskManager](#)
- package [com.sgi.sysadm.plugin](#)
- package [com.sgi.sysadm.ui](#)
- package [com.sgi.sysadm.ui.EditableList](#)
- package [com.sgi.sysadm.ui.HostContext](#)
- package [com.sgi.sysadm.ui.RButtonGroup](#)
- package [com.sgi.sysadm.ui.event](#)
- package [com.sgi.sysadm.ui.richText](#)
- package [com.sgi.sysadm.ui.taskData](#)
- package [com.sgi.sysadm.ui.treeView](#)
- package [com.sgi.sysadm.util](#)
- package [com.sgi.sysadm.util.SysUtil](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Index of all Fields and Methods

A

abort(). Static method in class com.sgi.sysadm.ui.HostContext
Exit all HostContexts in this JVM.

acceptsOperandType(String). Method in class com.sgi.sysadm.ui.TaskLoader
Determine if the Task accepts the given operand type.

activate(). Method in class com.sgi.sysadm.manager.RApplet
Called when the user activates (typically by clicking the mouse) the Applet.

ACTIVE_LINK_COLOR. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent
The resource *RichText.activeLinkColor* or *<componentName>.activeLinkColor* is an Integer specifying the color to be used in the text of a link when the user is holding the mouse button down over that link.

add(AbstractButton). Method in class com.sgi.sysadm.ui.RButtonGroup
Adds a button to the group.

ADD_BUTTON_LABEL. Static variable in class com.sgi.sysadm.ui.EditableList
The resource *EditableList.addButtonLabel* or *<componentName>.addButtonLabel* is the String displayed on the Add button.

ADD_MARGIN_HEIGHT. Static variable in interface com.sgi.sysadm.ui.ItemViewProperties
The property *ItemViewPanel.additionalInfo.marginHeight* is the height (in points) of the margin around the additional info panel.

ADD_MARGIN_WIDTH. Static variable in interface com.sgi.sysadm.ui.ItemViewProperties
The property *ItemViewPanel.additionalInfo.marginWidth* is the width (in points) of the margin around the additional info panel.

addActionListener(ActionListener). Method in class com.sgi.sysadm.ui.treeView.TreeViewPane
Add a listener that gets called when an action is performed on a node in the tree.

addAttrListener(AttrListener). Method in class com.sgi.sysadm.util.AttrBundle
Add a listener to get notified when Attributes are added, changed, or removed.

addBlinker(Blinker). Static method in class com.sgi.sysadm.ui.BlinkThread

addCategoryListener(CategoryListener). Method in class com.sgi.sysadm.category.Category
Convenience method to specify that the CategoryListener object is interested in receiving all notifications.

addCategoryListener(CategoryListener, NotificationFilter). Method in class com.sgi.sysadm.category.Category
Called by clients to add a CategoryListener to the list of objects which will receive types of notifications specified.

addComponent(Component). Method in class com.sgi.sysadm.ui.OneColumnPanel
Adds a component without a label to the panel.

addComponent(Component). Method in class com.sgi.sysadm.ui.TwoColumnPanel
Adds a component without a label to the panel

addComponent(Component, Component). Method in class com.sgi.sysadm.ui.OneColumnPanel
Adds a component and a label to the panel.

addComponent(Component, Component). Method in class com.sgi.sysadm.ui.TwoColumnPanel

Adds a component and a label to the panel.
addComponent(Component, String). Method in class com.sgi.sysadm.ui.LabelComponentPanel

Adds a component and a LinkLabel to a page.
addComponent(Component, String). Method in class com.sgi.sysadm.ui.OneColumnPanel

Adds a component and a label to the panel.
addComponent(Component, String). Method in class com.sgi.sysadm.ui.TwoColumnPanel

Adds a component and a label to the panel
addComponent(Component, String, Component). Method in class com.sgi.sysadm.ui.LabelComponentPanel

Adds a component and a label to the page.
addControlListener(TaskControlListener). Method in class com.sgi.sysadm.ui.TaskControlPanel

Register interest in TaskControlPanel events.
addEntry(String). Method in class com.sgi.sysadm.plugin.LogViewerFrame

addFocusListener(FocusListener). Method in class com.sgi.sysadm.ui.RTextField

Add a FocusListener to our list.
addIconListener(String, Object, int, int, RenderedObjectListener). Method in class com.sgi.sysadm.ui.IconRenderer

Adds a RenderedObjectListener to the renderer
addIconListener(String, Object, int, int, RenderedObjectListener). Method in class com.sgi.sysadm.ui.ResourceBasedIconRenderer

Adds a RenderedObjectListener to the renderer
addInTaskDataBinding(String, String). Method in class com.sgi.sysadm.ui.TaskLaunchComponent

Connects some parent's (ex., metatask) TaskData to the child's TaskData.
addItem(Item). Method in class com.sgi.sysadm.category.Category

Called by subclasses when a new Item is discovered at startup or when an Item is added.
addItemFinderListener(ItemFinderListener). Method in class com.sgi.sysadm.ui.ItemFinder

Add an ItemFinderListener to this ItemFinder
addItemListener(ItemListener). Method in class com.sgi.sysadm.ui.ItemTable

Add a ItemListener to receive ItemEvents when the Items are selected in the ItemTable, or when the selection changes.
addItemViewInfo(Component, Component). Method in class com.sgi.sysadm.ui.ItemViewPanel

Adds a label and value to the "info" part of the panel.
addItemViewInfo(Component, String). Method in class com.sgi.sysadm.ui.ItemViewPanel

Adds a label and value to the "info" part of the panel.
addItemViewInfo(String, Component). Method in class com.sgi.sysadm.ui.ItemViewPanel

Adds a label and value to the "info" part of the panel.
addItemViewInfo(String, String). Method in class com.sgi.sysadm.ui.ItemViewPanel

Adds a label and value to the "info" part of the panel.
addItemViewLaunchRequestListener(ItemViewLaunchRequestListener). Method in class com.sgi.sysadm.ui.ItemTable

Adds an ItemViewLaunchRequestListener that will be notified if a user requests that an Item View be launched (such as by clicking on a hyperlink in the ItemTable)
addItemViewLaunchRequestListener(ItemViewLaunchRequestListener). Method in class com.sgi.sysadm.ui.ItemTableController

Adds an ItemViewLaunchRequestListener.
addItemViewLaunchRequestListener(ItemViewLaunchRequestListener). Method in class com.sgi.sysadm.ui.ItemView

Adds an ItemViewLaunchRequestListener that will be notified if a user requests that a Item View be launched (such as be clicking on a hyperlink on the Item View)
addItemViewLaunchRequestListener(ItemViewLaunchRequestListener). Method in class com.sgi.sysadm.ui.ItemViewController

Adds a `ItemViewLaunchRequestListener` to the list of listeners that will be notified when a user clicks on a hyperlink to launch a new `ItemView`.

addItemViewLaunchRequestListener(`ItemViewLaunchRequestListener`). Method in class `com.sgi.sysadm.ui.ResultViewPanel`

Adds an `ItemViewLaunchRequestListener` to the list of listeners that will be notified if an `ItemView` launch is requested.

ADDITIONAL_INFO_RENDERER. Static variable in interface `com.sgi.sysadm.ui.ItemViewProperties`

A property `<name>.ItemView.additionalInfoRenderer` gives the fully qualified name of a class to use as the `ItemViewAdditionalInfoRenderer` for this `ItemView`.

addKeyCodeShortCut(`JButton`, `int`). Static method in class `com.sgi.sysadm.ui.UIContext`

Add an accelerator to a button using the specified `keyCode`.

addKeyCodeShortCut(`JButton`, `String`). Method in class `com.sgi.sysadm.ui.UIContext`

Add an accelerator to a button using the specified resource.

addLayoutComponent(`Component`, `Object`). Method in class `com.sgi.sysadm.ui.LinkPageLayout`

addLayoutComponent(`String`, `Component`). Method in class `com.sgi.sysadm.ui.LinkPageLayout`

addLinkListener(`LinkListener`). Method in class `com.sgi.sysadm.ui.richText.RichTextComponent`

Add a listener to be notified when the user clicks on a link.

addListSelectionListener(`ListSelectionListener`). Method in class `com.sgi.sysadm.ui.ItemTablePanel`

Adds a `ListSelectionListener` to the table's selection model

addModule(`String`). Static method in class `com.sgi.sysadm.util.Log`

Add a module to the set of modules to display.

addNameListener(`String`, `Object`, `RenderedObjectListener`). Method in class `com.sgi.sysadm.ui.NameRenderer`

Adds a `RenderedObjectListener` to the renderer

addNotify(). Method in class `com.sgi.sysadm.ui.ComponentTable`

Override `addNotify()` in order to initialize table headers.

addNotify(). Method in class `com.sgi.sysadm.ui.EditableList`

Called when we're added to a `Container`.

addOptionalTaskComponent(`Component`). Method in class `com.sgi.sysadm.ui.TaskPage`

Adds an optional component but no label to the `TaskPage`

addOptionalTaskComponent(`Component`, `Component`). Method in class `com.sgi.sysadm.ui.TaskPage`

Adds an optional component and a label to the `TaskPage`.

addOptionalTaskComponent(`Component`, `String`). Method in class `com.sgi.sysadm.ui.TaskPage`

Adds a component and a label to the `TaskPage`.

addOptionalTaskComponent(`Component`, `String`, `Component`). Method in class `com.sgi.sysadm.ui.TaskPage`

Adds an optional component and a label to the `TaskPage`.

addOutTaskDataBinding(`String`, `String`). Method in class `com.sgi.sysadm.ui.TaskLaunchComponent`

Connects some child's `TaskData` to the parent's (ex., metatask) `TaskData`.

addPrivs(`String[]`, `String`, `ResultListener`). Method in interface `com.sgi.sysadm.util.PrivBroker`

Add a set of privileges for the user.

addProcessListener(`ProcessListener`). Method in class `com.sgi.sysadm.util.ProcessWatcher`

Add a listener interested in `ProcessEvents`.

addRButtonGroupListener(`RButtonGroup`, `RButtonGroupListener`). Method in class `com.sgi.sysadm.ui.RButtonGroup`

Add a `RButtonGroupListener` to this `RButtonGroup`.

addRenderedObjectListener(`String`, `RenderedObjectListener`, `Object`). Method in class `com.sgi.sysadm.ui.GenericItemRenderer`

Adds a `RenderedObjectListener` to the renderer

addSpanningComponent(`Component`). Method in class `com.sgi.sysadm.ui.TwoColumnPanel`

Adds a component that spans both the left and right columns.

addTableSortRequestListener(TableSortRequestListener). Method in class com.sgi.sysadm.ui.ComponentTable
 Adds a listener that will be notified when the user requests that a table be sorted on a particular column.

addTableSortRequestListener(TableSortRequestListener). Method in class com.sgi.sysadm.ui.ItemTablePanel
 Adds a TableSortRequestListener to this table.

addTaskComponent(Component). Method in class com.sgi.sysadm.ui.TaskPage
 Adds a component but no label to the TaskPage

addTaskComponent(Component, Component). Method in class com.sgi.sysadm.ui.TaskPage
 Adds a component and a label to the TaskPage.

addTaskComponent(Component, String). Method in class com.sgi.sysadm.ui.TaskPage
 Adds a component and a label to the TaskPage.

addTaskComponent(Component, String, Component). Method in class com.sgi.sysadm.ui.TaskPage
 Adds a component and a label to the TaskPage.

addTaskDataBinder(String, TaskDataBinder). Method in class com.sgi.sysadm.ui.taskData.TaskData
 Add a TaskDataBinder that gets notified when a particular Attribute changes.

addTaskDoneListener(TaskDoneListener). Method in class com.sgi.sysadm.ui.Task
 Register interest in task completion.

addTaskDoneListener(TaskDoneListener). Method in class com.sgi.sysadm.ui.TaskFrame
 Register interest in task completion.

addTaskLaunchComponentListener(TaskLaunchComponentListener). Method in class com.sgi.sysadm.ui.TaskLaunchComponent
 Adds a listener to the list of listeners that will be notified when this TaskLaunchComponent changes state.

addTaskLaunchRequestListener(TaskLaunchRequestListener). Method in class com.sgi.sysadm.ui.ItemView
 Adds a TaskLaunchRequestListener that will be notified if a user requests that a Task be launched (such as by clicking in a TaskShelf contained by this ItemView)

addTaskLaunchRequestListener(TaskLaunchRequestListener). Method in class com.sgi.sysadm.ui.ResultViewPanel
 Adds a TaskLaunchRequestListener to the list of listeners that will be notified if an Task launch is requested.

addTaskLaunchRequestListener(TaskLaunchRequestListener). Method in class com.sgi.sysadm.ui.TaskShelfPanel
 Adds a listener to the list that will be notified when a user requests that a task be launched.

addTitleListener(TitleListener). Method in class com.sgi.sysadm.ui.ItemTable
 Adds a TitleListener to the ItemTable.

addTitleListener(TitleListener). Method in class com.sgi.sysadm.ui.ItemView
 Adds a title Listener

addTitleListener(TitleListener). Method in class com.sgi.sysadm.ui.Task
 Register interest in changes to the Task title.

addTitleListener(TitleListener). Method in class com.sgi.sysadm.ui.TaskContext
 Add a listener to the list of those interested in changes to the Task title.

addTreeSelectionListener(TreeSelectionListener). Method in class com.sgi.sysadm.ui.treeView.TreeViewPane
 Adds a listener for TreeSelection events.

ALIGNMENT. Static variable in interface com.sgi.sysadm.ui.ItemTableProperties
 A property <name>.ItemTable.alignment.<column> specifies the alignment to use for a column.

ALL_DATA_VERIFIER. Static variable in class com.sgi.sysadm.ui.TaskContext

When the user presses the OK button, the Task will call `TaskContext.allDataOK` (`TaskDataVerifiers.MUST_BE_SET`).

ALL_ITEMS. Static variable in class `com.sgi.sysadm.category.NotificationFilter`
`NotificationFilter` specifying interest in all Item(s) of a Category.

ALL_OPERAND_TYPES. Static variable in class `com.sgi.sysadm.ui.Task`
The String `Task.allOperandTypes` is the value a Task should use for the property `Task.operandTypeAccepted` if the Task will accept an operand of any type.

allDataOK(int, Object, ResultListener). Method in class `com.sgi.sysadm.ui.TaskContext`
Verify that all TaskData is valid.

alreadyExists(EditableList). Static method in class `com.sgi.sysadm.ui.EditableList.DefaultEditVerifier`
Checks to see if there is already a row in the list has the same values for each column as the values in the editor.

ALWAYS. Static variable in class `com.sgi.sysadm.ui.LabelComponentConstraints`
Always attach the component the the right hand side

ALWAYS_CLEAR. Static variable in class `com.sgi.sysadm.ui.ItemFinder`
Pass this value to `setClearBehavior` to specify that the ItemFinder should clear the text when the Category is set via the `setCategory` method.

appendPage(GuidePage). Method in class `com.sgi.sysadm.ui.Guide`
Append a GuidePage to the ordered list of GuidePages.

appendTaskDataVerifier(String, TaskDataVerifier). Method in class `com.sgi.sysadm.ui.TaskContext`
Append a TaskDataVerifier onto the list of verifiers.

AppExitHandler(). Constructor for class `com.sgi.sysadm.ui.HostContext.AppExitHandler`

apply(Object, Object). Method in interface `com.sgi.sysadm.util.BinaryPredicate`
Tests whether or not the arguments satisfy some condition.

ArrowIcon(ResourceStack, int). Constructor for class `com.sgi.sysadm.ui.ArrowIcon`
Constructor

ArrowIcon(ResourceStack, String, int). Constructor for class `com.sgi.sysadm.ui.ArrowIcon`
Constructor

ASSERT. Static variable in class `com.sgi.sysadm.util.Log`
Used with `leveOn()/levelOff()` to enable/disable display of messages from the `assert()` logging method.

assert(boolean, String). Static method in class `com.sgi.sysadm.util.Log`
Verify some condition is true; if not log a message and terminate the program.

Association(String, String, String). Constructor for class `com.sgi.sysadm.category.Association`
Association constructor.

AssociationItemFinderBinder(ItemFinder, String, String, HostContext). Constructor for class `com.sgi.sysadm.ui.taskData.AssociationItemFinderBinder`
Construct a AssociationItemFinderBinder.

attachBottom. Variable in class `com.sgi.sysadm.ui.LabelComponentConstraints`
The bottommost Component in the layout will be attached to the bottom of the Container if `attachBottom` is set to true for that component.

attrAdded(AttrEvent). Method in interface `com.sgi.sysadm.util.AttrListener`
Called when an Attribute is added to a AttrBundle.

attrAdded(AttrEvent). Method in class `com.sgi.sysadm.category.CategoryAdapter`

attrAdded(AttrEvent). Method in class `com.sgi.sysadm.ui.taskData.TaskDataBinder`
Called whenever an Attribute in TaskData is added.

AttrBundle(). Constructor for class `com.sgi.sysadm.util.AttrBundle`
Construct an AttrBundle with empty type and selector.

AttrBundle(AttrBundle). Constructor for class `com.sgi.sysadm.util.AttrBundle`
Construct an AttrBundle that is a copy of other.

AttrBundle(String). Constructor for class `com.sgi.sysadm.util.AttrBundle`

Construct an AttrBundle from serialized format.

AttrBundle(String, String). Constructor for class com.sgi.sysadm.util.AttrBundle
Construct an AttrBundle with type and selector.

attrChanged(AttrEvent). Method in interface com.sgi.sysadm.util.AttrListener
Called when an Attribute within an AttrBundle changes.

attrChanged(AttrEvent). Method in class com.sgi.sysadm.category.CategoryAdapter

attrChanged(AttrEvent). Method in class com.sgi.sysadm.ui.taskData.TaskDataBinder
Called whenever an Attribute in TaskData changes.

AttrEvent(AttrBundle, Attribute). Constructor for class com.sgi.sysadm.util.AttrEvent
Construct an AttrEvent.

Attribute(String, AttrBundle). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute of type AttrBundle

Attribute(String, boolean). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute of type boolean.

Attribute(String, double). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute of type double.

Attribute(String, long). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute of type long.

Attribute(String, Object). Constructor for class com.sgi.sysadm.util.Attribute
Construct an attribute from a name and a value object

Attribute(String, String). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute of type String.

Attribute(String, String, String). Constructor for class com.sgi.sysadm.util.Attribute
Construct an Attribute from String representations of type and value.

AttributeAbstractButtonBinder(AbstractButton, Attribute). Constructor for class com.sgi.sysadm.ui.taskData.AttributeAbstractButtonBinder
Construct an AttributeAbstractButtonBinder.

attrRemoved(AttrEvent). Method in interface com.sgi.sysadm.util.AttrListener
Called when an Attribute is removed from a AttrBundle.

attrRemoved(AttrEvent). Method in class com.sgi.sysadm.category.CategoryAdapter

attrRemoved(AttrEvent). Method in class com.sgi.sysadm.ui.taskData.TaskDataBinder
Called whenever an Attribute in TaskData is removed.

AUTO_WRAP. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent
The resource *RichText.autoWrap* or *<componentName>.autoWrap* is a Boolean specifying whether RichText should wrap long lines in paragraphs.

B

BACKGROUND. Static variable in interface com.sgi.sysadm.ui.treeView.TreeViewProperties
The property *<prefix>.background* specifies the color to be used as the background for the TreeViewPane.

BAD_PAGE_TYPE. Static variable in interface com.sgi.sysadm.manager.taskManager.TaskManagerProperties
The property *TaskManager.Error.badPageType* is a format String that is displayed if a table of contents page plugin is not of the expected type (specifically TaskManagerPanel).

BASED_ON. Static variable in interface com.sgi.sysadm.ui.ItemTableProperties
A property *<name>.ItemTable.basedOn.<column>* is a string that tells which Attribute of the Item is associated with the column given by *<column>*.

BASED_ON. Static variable in interface com.sgi.sysadm.ui.ItemViewProperties

A property <name>.*ItemView.basedOn*.<field> is a string that tells which Attribute of the Item is associated with the field given by <field>, where <name> is the name that was passed the the constructor of ItemViewController.

BEEP. Static variable in class com.sgi.sysadm.ui.FilteredTextField

Passing in the special character FilteredTextField.BEEP as the designated replacement character will cause the text field to beep instead of converting disallowed characters.

beginBlockChanges(). Method in class com.sgi.sysadm.category.Category

Begin a change block.

beginBlockChanges(). Method in class com.sgi.sysadm.category.CategoryAdapter

beginBlockChanges(). Method in interface com.sgi.sysadm.category.CategoryListener

Called by Category prior to a block of changes.

bgnclosedline(). Method in class com.sgi.sysadm.ui.FtrIcon

Start drawing a series of closed lines.

bgnline(). Method in class com.sgi.sysadm.ui.FtrIcon

Start drawing a series of connected lines.

bgnoutlinepolygon(). Method in class com.sgi.sysadm.ui.FtrIcon

Start drawing an outlined polygon.

bgnpoint(). Method in class com.sgi.sysadm.ui.FtrIcon

Start drawing a series of points.

bgnpolygon(). Method in class com.sgi.sysadm.ui.FtrIcon

Start drawing a polygon.

bind(TaskData, String, AbstractButton). Static method in class

com.sgi.sysadm.ui.taskData.BooleanAbstractButtonBinder

Bind the Attribute name in taskData to button, so that when button changes the Attribute is updated and vice versa.

bind(TaskData, String, AbstractButton, Attribute). Static method in class

com.sgi.sysadm.ui.taskData.AttributeAbstractButtonBinder

Bind a Button to a piece of TaskData.

bind(TaskData, String, AbstractButton, boolean). Static method in class

com.sgi.sysadm.ui.taskData.BooleanAbstractButtonBinder

Bind the Attribute name in taskData to button, so that when button changes the Attribute is updated and vice versa.

bind(TaskData, String, Component). Static method in class

com.sgi.sysadm.ui.taskData.BooleanComponentEnabledBinder

Bind the Attribute name in taskData to comp so that when the Attribute changes the enabled state of comp changes.

bind(TaskData, String, Component). Static method in class

com.sgi.sysadm.ui.taskData.StringComponentEnabledBinder

Bind component to the Attribute name in taskData so that when the Attribute changes the component is enabled/disabled.

bind(TaskData, String, ItemFinder). Static method in class

com.sgi.sysadm.ui.taskData.SelectorItemFinderBinder

Bind the Attribute name in taskData to finder, so that when finder changes the Attribute is updated and vice versa.

bind(TaskData, String, ItemFinder). Static method in class

com.sgi.sysadm.ui.taskData.TextItemFinderBinder

Bind the Attribute name in taskData to finder, so that when finder changes the Attribute is updated and vice versa.

bind(TaskData, String, ItemFinder, String, String). Static method in class

com.sgi.sysadm.ui.taskData.ReasonItemFinderBinder

Bind the Attribute name in taskData to finder, so that when finder changes the Attribute is updated and vice versa.

bind(TaskData, String, JComboBox, Object, long). Static method in class com.sgi.sysadm.ui.taskData.LongJComboBoxBinder

Bind the Attribute name in taskData to box, so that when item is selected in box the Attribute is changed to value, and vice versa.

bind(TaskData, String, JComboBox, Object, String). Static method in class com.sgi.sysadm.ui.taskData.StringJComboBoxBinder

Bind the Attribute name in taskData to box, so that when entry is selected in box the Attribute is changed to value, and vice versa.

bind(TaskData, String, JLabel). Static method in class com.sgi.sysadm.ui.taskData.StringJLabelBinder
Bind label to the Attribute name in taskData so that when the Attribute changes label is updated.

bind(TaskData, String, JTextComponent). Static method in class com.sgi.sysadm.ui.taskData.LongJTextComponentBinder

Bind the Attribute name in taskData to text, so that when text changes the Attribute is updated and vice versa.

bind(TaskData, String, JTextComponent). Static method in class com.sgi.sysadm.ui.taskData.StringJTextComponentBinder

Bind the Attribute name in taskData to text, so that when text changes the Attribute is updated and vice versa.

bind(TaskData, String, RButtonGroup). Static method in class com.sgi.sysadm.ui.taskData.LongRButtonGroupBinder

Bind the Attribute name in taskData to group, so that when a button is selected in group the Attribute is changed to match, and vice versa.

bind(TaskData, String, String, String, HostContext, ItemFinder). Static method in class com.sgi.sysadm.ui.taskData.AssociationItemFinderBinder

Bind itemfinder to the Attribute parentSelectorKey in taskData so that when the Attribute changes an association is updated.

bind(TaskData, String, TaskLaunchComponent). Static method in class com.sgi.sysadm.ui.taskData.StringTaskLaunchComponentBinder

Bind tlc to the Attribute name in taskData so that when the Attribute changes the status of the TaskLaunchComponent is set

BlankIcon(int, int). Constructor for class com.sgi.sysadm.ui.BlankIcon
Constructor.

BLINK_ATTRS. Static variable in class com.sgi.sysadm.ui.ResourceBasedIconRenderer

A resource <name>.blinkAttr is a string array that gives the names of Attributes in the Item.

BLINK_VALUES. Static variable in class com.sgi.sysadm.ui.ResourceBasedIconRenderer

A resource <name>.blinkValue is a string array that gives the values of Attributes that an Item may have.

blinkOff(). Method in interface com.sgi.sysadm.ui.Blinker
Called when the icon should blink off

blinkOn(). Method in interface com.sgi.sysadm.ui.Blinker
Called when the icon should blink on

BlinkThread(). Constructor for class com.sgi.sysadm.ui.BlinkThread

blockAllClients(Window). Method in class com.sgi.sysadm.ui.HostContext

Iterate through all of the clients and block input to them by raising their glass pane.

blockInput(boolean). Method in class com.sgi.sysadm.ui.UIContext

Prevent (or allow) input to the dialogParent by raising (or lowering) the glassPane.

blockInput(boolean, ResultListener). Method in class com.sgi.sysadm.ui.UIContext

Prevent (or allow) input to the dialogParent by raising (or lowering) the glassPane, notifying

listener when input has been blocked (or unblocked).

BOLD_LINKS. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent

The resource *RichText.boldLinks* or *<componentName>.boldLinks* is a Boolean specifying whether or not links should always be bold.

BooleanAbstractButtonBinder(AbstractButton). Constructor for class

com.sgi.sysadm.ui.taskData.BooleanAbstractButtonBinder

Construct a BooleanAbstractButtonBinder.

BooleanAbstractButtonBinder(AbstractButton, boolean). Constructor for class

com.sgi.sysadm.ui.taskData.BooleanAbstractButtonBinder

Construct a BooleanAbstractButtonBinder.

BooleanComponentEnabledBinder(Component). Constructor for class

com.sgi.sysadm.ui.taskData.BooleanComponentEnabledBinder

Construct a BooleanComponentEnabledBinder.

booleanValue(). Method in class com.sgi.sysadm.util.Attribute

Get the value of this Attribute as a boolean.

BOTTOM. Static variable in class com.sgi.sysadm.ui.LabelComponentPanel

A resource *<component name | component's class name |*

"LabelComponentPanel">.attachBottom determines whether the bottom-most component in the layout is stretched to touch the bottom of the panel.

BOTTOM. Static variable in class com.sgi.sysadm.ui.TaskLaunchComponent

The resource *TaskLaunchComponent.margin.bottom* gives the number of points to use for the bottom of the margin inside the component.

BOTTOM_GAP. Static variable in class com.sgi.sysadm.ui.LabelComponentPanel

A resource *<component name | component's class name | "LabelComponentPanel">.bottomGap* determines the vertical gap (in points) between one component and the next.

BOTTOM_INSET. Static variable in class com.sgi.sysadm.ui.Task

The property *Task.bottomInset* is an integer that defines the inset, in points, between the bottom of the task container and the task interface.

BOTTOM_INSET. Static variable in class com.sgi.sysadm.ui.UIContext

The property *UIContext.Dialog.bottomInset* is an integer that defines the height, in points, of the inset between the bottom of a dialog and the contents of a dialog.

BOTTOM_MARGIN. Static variable in class com.sgi.sysadm.ui.RPasswordField

A resource *Field.bottomMargin* defined in com.sgi.sysadm.ui.SysadmUI.properties that specifies the bottom margin to be used in Task TextFields.

BOTTOM_MARGIN. Static variable in class com.sgi.sysadm.ui.RTextField

A resource *Field.bottomMargin* defined in com.sgi.sysadm.ui.SysadmUI.properties that specifies the amount of space between the bottom of a letter descender (ex., the tail of a "g"), and the border of the text field.

bottomGap. Variable in class com.sgi.sysadm.ui.LabelComponentConstraints

The number of pixels to put below each component in the layout and the label or component directly below it.

BULLET_LEFT_OFFSET. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent

The resource *RichText.bulletLeftOffset* or *<componentName>.bulletLeftOffset* is an Integer specifying the left offset of a bullet in a list item relative to the paragraph it is in.

BULLET_TOP_OFFSET. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent

The resource *RichText.bulletTopOffset* or *<componentName>.bulletTopOffset* is an Integer specifying padding at the top of a bullet.

BUNDLE_SUFFIX. Static variable in class com.sgi.sysadm.util.ResourceStack

The suffix to append to all property files For example, myProperties<BUNDLE_SUFFIX>.properties.

bundleValue(). Method in class com.sgi.sysadm.util.Attribute

Get the value of this Attribute as an `AttrBundle`.

busy(). Method in class `com.sgi.sysadm.ui.UIContext`
Set the cursor to a busy cursor and do not allow user input.

busy(ResultListener). Method in class `com.sgi.sysadm.ui.UIContext`
Set the cursor to a busy cursor and do not allow user input; notify the caller when the cursor has changed to busy.

busy(String). Method in class `com.sgi.sysadm.ui.UIContext`
Post a modal busy dialog with a localized busy message and do not allow user input to the dialog parent.

busy(String, ActionListener). Method in class `com.sgi.sysadm.ui.UIContext`
Post a modal busy dialog with a localized busy message and a cancel button; do not allow user input to the dialog parent.

BUSY_ICON. Static variable in class `com.sgi.sysadm.ui.UIContext`
The property `UIContext.Dialog.busyIcon` is the CLASSPATH relative name of the icon image file of the icon to use in the busy dialog.

BUTTON_ARROW_SPACING. Static variable in class `com.sgi.sysadm.ui.TaskControlPanel`
The property `TaskControlPanel.buttonArrowSpacing` is the spacing in points between the arrow icon and the text in the "Next" and "Prev" buttons.

BUTTON_ITEM. Static variable in interface `com.sgi.sysadm.manager.taskManager.TaskManagerProperties`
The property set `TaskManager.buttonItem<n>` defines the labels to use for the product-specific buttons to add to the bottom of the TaskManager window.

BUTTON_SPACING. Static variable in interface `com.sgi.sysadm.manager.taskManager.TaskManagerProperties`
The property `TaskManager.buttonSpacing` is an integer which defines the amount of space, in points, between the buttons at the bottom of the TaskManager window.

BUTTON_TARGET. Static variable in interface `com.sgi.sysadm.manager.taskManager.TaskManagerProperties`
The String `.target`, when appended to `BUTTON_ITEM`, becomes a property that describes the class name to load when the button is pressed.

C

CANCEL_BUTTON_KEY_CODE. Static variable in class `com.sgi.sysadm.ui.TaskControlPanel`
The property `TaskControlPanel.cancelButtonKeyCode` is the key code of the accelerator for the "Cancel" button.

CANCEL_BUTTON_LABEL. Static variable in class `com.sgi.sysadm.ui.TaskControlPanel`
The property `TaskControlPanel.cancelButtonLabel` is a String that is displayed on the cancel button.

CANCEL_LABEL. Static variable in class `com.sgi.sysadm.ui.UIContext`
The property `UIContext.Dialog.cancelLabel` is a string displayed on the cancel button of the busy dialog.

CANCELED. Static variable in class `com.sgi.sysadm.ui.TaskLaunchComponent`
A resource `TaskLaunchComponent.canceled` is the string to display for the status if the Task has been canceled.

CANCELED. Static variable in class `com.sgi.sysadm.ui.TaskLaunchComponentState`
The state that indicates that the Task has been canceled

CANCELLED. Static variable in class `com.sgi.sysadm.ui.event.TaskResult`

Results of type `TaskResult.CANCELLED` are generated when a task is cancelled by the user.

CANT_LOAD_CLASS. Static variable in interface `com.sgi.sysadm.ui.ItemTableProperties`
 A property `ItemTable.Error.cantLoadClass` is the error message that will be displayed if a necessary class can't be loaded.

CANT_LOAD_CLASS. Static variable in interface `com.sgi.sysadm.ui.ItemViewProperties`
 A property `ItemView.Error.cantLoadClass` gives the error message that will be displayed if a necessary class can't be loaded.

CATEGORY. Static variable in class `com.sgi.sysadm.category.Category`
 This is the suffix that all Category selectors must end with.

CATEGORY. Static variable in interface `com.sgi.sysadm.ui.ItemTableProperties`
 A property `<name>.ItemTable.category.<column>` is a string specifying the Category to use to launch the ItemView for a particular column.

CATEGORY. Static variable in interface `com.sgi.sysadm.ui.ItemViewProperties`
 A property `<name>.ItemView.category.<field>` is needed only if `<field>` is using `richText` as its display method.

CATEGORY. Static variable in class `com.sgi.sysadm.ui.ResourceBasedNameRenderer`
 A resource `<category name>.categoryName` is a string that specifies the user visible name of the Category.

Category(String). Constructor for class `com.sgi.sysadm.category.Category`
 Category constructor.

CATEGORY_ONLY. Static variable in class `com.sgi.sysadm.ui.GenericItemRenderer`
 A string that is passed as `itemSelector` to `addRenderedObjectListener` and `removeRenderedObjectListener` to signal that the caller is requesting that the rendered object contain information about the Category in general, instead of a particular Item.

CATEGORY_PLURAL. Static variable in class `com.sgi.sysadm.ui.ResourceBasedNameRenderer`
 A resource `<category name>.pluralCategoryName` is a string that specifies the user visible name of the Category in its plural form.

CategoryAdapter(). Constructor for class `com.sgi.sysadm.category.CategoryAdapter`

Cell(). Constructor for class `com.sgi.sysadm.ui.EditableList.Cell`
 Construct a cell specifying placement at row 0 and column 0.

Cell(int, int). Constructor for class `com.sgi.sysadm.ui.EditableList.Cell`
 Construct a Cell.

Cell(int, int, int, int). Constructor for class `com.sgi.sysadm.ui.EditableList.Cell`
 Construct a Cell.

CELL_BORDER_HEIGHT. Static variable in interface `com.sgi.sysadm.ui.treeView.TreeViewProperties`
 The property `<prefix>.cellBorderHeight` specifies the height, in points, of the border around each Item in the tree.

CELL_BORDER_WIDTH. Static variable in interface `com.sgi.sysadm.ui.treeView.TreeViewProperties`
 The property `<prefix>.cellBorderWidth` specifies the width, in points, of the border around each Item in the tree.

CHANGED_SINCE_LAST_ADD_ATTR_NAME. Static variable in class `com.sgi.sysadm.ui.EditableList`
 The resource `EditableList.changedSinceLastAddAttrName` or `<componentName>.changedSinceLastAddAttrName` specifies the name of a Boolean Attribute in TaskData that gets bound to the state of the editor: the Attribute will be `true` if the user has made changes and `false` otherwise.

changedSinceLastAdd(). Method in class `com.sgi.sysadm.ui.EditableList`
 Determine whether the user has changed anything in the editor since the last time the "Add" button was pressed.

changedSinceLastAdd(EditableList). Method in class com.sgi.sysadm.ui.EditableList.DefaultEditVerifier

Called to determine whether user has made any changes.

changedSinceLastAdd(EditableList). Method in interface com.sgi.sysadm.ui.EditableList.EditVerifier

Called by EditableList.changedSinceLastAdd() to determine whether the user had made any changes.

changedUpdate(DocumentEvent). Method in class com.sgi.sysadm.ui.taskData.LongJTextComponentBinder

Called when our JTextComponent changes.

changedUpdate(DocumentEvent). Method in class com.sgi.sysadm.ui.taskData.StringJTextComponentBinder

Called when our JTextComponent changes.

changeItem(Item). Method in class com.sgi.sysadm.category.Category

Called by subclasses when an item in the system changed.

checkPassword(ResultListener). Method in interface com.sgi.sysadm.util.PrivBroker

Check that the current password (set via setPassword) is valid.

checkPrivs(ResultListener). Method in class com.sgi.sysadm.ui.Task

checkPrivs() is a service provided by the base class for checking and obtaining the privileges needed to perform the Task.

checkPrivs(String[], ResultListener). Method in interface com.sgi.sysadm.util.PrivBroker

Check to see if we have a set of privileges.

checkPrivs(TaskLoader[], ResultListener). Method in class com.sgi.sysadm.ui.Task

Checks all privileges needed for several tasks.

CHILD_FONT. Static variable in interface com.sgi.sysadm.ui.treeView.TreeViewProperties

The property <prefix>.childFont specifies the name of the font to be used to display the name of all Items in the tree (except for the Item at the root of the tree).

CLASS. Static variable in interface com.sgi.sysadm.manager.taskManager.TaskManagerProperties

If a Table of Contents item has the type *class*, it will display a TaskManagerPanel plugin in the DisplayArea of the TaskManager window.

CLASS_NAME. Static variable in class com.sgi.sysadm.ui.richText.RichTextComponent

RichText is the name prepended to resource names when looking for resource settings that apply to all instances.

CLASS_PAGE_TYPE. Static variable in class com.sgi.sysadm.manager.taskManager.DisplayPage

Use this value as an argument to DisplayPage.setType() if the page to display is a TaskManagerPanel.

ClassLoadException(String). Constructor for class com.sgi.sysadm.util.SysUtil.ClassLoadException

A constructor that takes a className.

ClassLoadException(String, String). Constructor for class com.sgi.sysadm.util.SysUtil.ClassLoadException

A constructor that takes a className and errorString

CLEAR_BUTTON_LABEL. Static variable in class com.sgi.sysadm.manager.taskManager.SearchPanel

The property *SearchPanel.clearButtonLabel* is the String to display on the button the user presses to clear the keyword field.

CLEAR_IF_SWITCHING. Static variable in class com.sgi.sysadm.ui.ItemFinder

Pass this value to setClearBehavior to specify that the ItemFinder should clear the text field only if it used to be displaying a Category, and setCategory was called again.

clearList(). Method in class com.sgi.sysadm.ui.EditableList

Removes all the row data from the list.

Client(Frame, UIContext). Constructor for class com.sgi.sysadm.ui.HostContext.Client
Constructor.

clone(). Method in class com.sgi.sysadm.util.AttrBundle
Create a new AttrBundle that is a copy of this AttrBundle.

clone(). Method in class com.sgi.sysadm.category.Item
Get an object that is a copy of this one.

clone(). Method in class com.sgi.sysadm.ui.LabelComponentConstraints
Make a copy of this object.

clone(). Method in class com.sgi.sysadm.category.NotificationFilter
Copy the value of this object

clone(). Method in class com.sgi.sysadm.util.ResourceStack
Makes an exact copy of the ResourceStack.

CLOSE_BUTTON_LABEL. Static variable in interface com.sgi.sysadm.manager.taskManager.TaskManagerProperties
The property *TaskManager.closeButtonLabel* is a string that will be displayed on the right-most button at the bottom of the TaskManager window.

CLOSED_ICON. Static variable in interface com.sgi.sysadm.ui.treeView.TreeViewProperties
The property <prefix>.closedIcon specifies the package-qualified name of the icon to display when a node in the tree has children and those children are not visible, that is, when the node is closed.

collapseAll(). Method in class com.sgi.sysadm.ui.treeView.TreeViewPane
Collapse all of the nodes in the tree.

COLOR. Static variable in class com.sgi.sysadm.ui.RLabel
A resource <name>.color is a resource that specifies the color to use for the RLabel, where <name> is the name passed to the constructor.

color(int). Method in class com.sgi.sysadm.ui.FtrIcon
Set the current color.

color(Object). Method in class com.sgi.sysadm.ui.FtrIcon
Set the current color.

COLUMN_ATTR_NAME. Static variable in class com.sgi.sysadm.ui.EditableList
The resource set *EditableList.columnAttrName*<n> or <componentName>.columnAttrName<n> specifies the Attribute name for each column in the list.

COLUMN_RENDERER. Static variable in interface com.sgi.sysadm.ui.ItemTableProperties
A property <name>.ItemTable.columnRenderer gives the fully qualified name of a class to use as the ItemTableColumnRenderer for this ItemTable.

COLUMN_WIDTH. Static variable in class com.sgi.sysadm.ui.EditableList
The resource set *EditableList.columnWidth*<n> or <componentName>.columnWidth<n> specifies the widths in points of the columns in the list.

COLUMNS. Static variable in interface com.sgi.sysadm.ui.ItemTableProperties
A set of properties *ItemTable.column*<n> name the columns that will be displayed in the ItemTable, and define the default order.

compareItems(Item, Item). Method in interface com.sgi.sysadm.category.ItemComparator
Compare two Item(s) for the purpose of sorting.

compareItemsForItemTable(Item, Item, String). Method in interface com.sgi.sysadm.ui.ItemTableColumnRenderer
Compare two items based on a particular column.

COMPLETED. Static variable in class com.sgi.sysadm.ui.TaskLaunchComponent
A resource *TaskLaunchComponent.completed* is the string to display for the status if the Task has been successfully completed.

COMPONENT_VERTICAL_SPACING. Static variable in class com.sgi.sysadm.ui.OneColumnPanel
A resource <name>.componentVerticalSpacing is an integer that gives the vertical space (in points) to put between the components.

ComponentTable(). Constructor for class com.sgi.sysadm.ui.ComponentTable

ComponentTable(int, int). Constructor for class com.sgi.sysadm.ui.ComponentTable
ComponentTable(Object[][], Object[]). Constructor for class com.sgi.sysadm.ui.ComponentTable
ComponentTable(TableModel). Constructor for class com.sgi.sysadm.ui.ComponentTable
ComponentTable(TableModel, TableColumnModel). Constructor for class com.sgi.sysadm.ui.ComponentTable
ComponentTable(TableModel, TableColumnModel, ListSelectionModel). Constructor for class com.sgi.sysadm.ui.ComponentTable
ComponentTable(Vector, Vector). Constructor for class com.sgi.sysadm.ui.ComponentTable
computeProductAttrs(String). Method in class com.sgi.sysadm.ui.ProductInfo
 Subclasses override this method that ProductInfo calls when it needs the product attributes for a particular product.
computeSelector(String, String, String). Static method in class com.sgi.sysadm.category.Association
 Compute the selector for this Association
computeTaskList(String). Method in class com.sgi.sysadm.ui.TaskRegistry
 Subclasses override this method that TaskRegistry calls when it needs the list of tasks for a particular category.
createFrame(). Method in class com.sgi.sysadm.manager.RApp
 Called by initApp() to initiate creation of the application frame.
createFrame(). Method in class com.sgi.sysadm.manager.RunItemTable
createFrame(). Method in class com.sgi.sysadm.manager.RunItemView
cr

▼ Rhino

[Overview](#)[News](#)[Documentation](#)[FAQ](#)[Mailing List](#)[How To Contribute](#)[To-Do List](#)[Download](#)[License](#)

▼ SGI Open Source

[Project List](#)

▼ SGI Support and Resources

[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)

▼ Community Resources

[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)

The Rhino To-Do List

These are specific known problems and which should be fixed in Rhino. [Feedback](#) and patches w gratefully accepted!

Known Bugs

See also [open Rhino bugs](#) in [Bugzilla](#).

If nobody's name is next to a bug, either no one's working on it, or Rusty is working on it. Frequent two states are indistinguishable.

- **i18n is broken** in the sysadmd protocol. (The client side is localized, and the server side localized, but at the moment there's no way for the client to tell the server what locale to use.)
- **readlog is missing.** This privileged command displays portions of the system log(s), but present on Linux yet because it used `_getpty()` on IRIX. (Aseel Abbas fixed this; it will prob be in 1.3.8.)
- In `build/rpm/rpm.spec.in`, the package groups are not quite right, and the descriptions could be better. Also, there have been [some reports](#) that **our RPM spec file is an abomination** (a should not be named "rpm.spec").

Missing Features

- The **ssh support** should be turned back on. It was removed for legal (export) reasons; we to determine whether those reasons are still relevant. (Inside SGI, this is bug [797618](#). wes knows about this.)
- There needs to be a **Privilege Manager GUI** for administering privileges. It would be really simple, and would make a great demo. (And it would be useful!)
- The **i18n licensing mess** needs to be resolved. Currently Rhino uses GNU gettext on platforms which support it (such as Linux), and no i18n on platforms which don't (such as IRIX). We to find out whether there's a way to use catgets through gettext or use the included libintl (which provides an implementation of gettext for platforms which don't support it) while remaining LGPL. If we can't do this, we need to scrap gettext and use catgets instead.

Originally Rhino used an i18n API which was unavailable on Linux. It was converted to use GNU gettext; unfortunately, both the code provided with gettext for encapsulating catgets and the gettext implementation licensed under the GPL, so we can't use them and be LGPL. Rusty sent mail to Ulrich Drepper some time ago, never received a reply, so this needs some more investigation.

- **mkrhinoism improvements:**
 - It ought to let you choose whether to create an entire TaskManager, Category, an of Tasks, or whether to create a single Task for inclusion in an existing project. On other hand, if you've got an existing project, it's probably easier to just copy one of existing tasks.)
 - It ought to let you choose whether to use GNU or SGI build trees.
 - It ought to let you choose what kind of license you want in the generated code (GPL, LGPL, other), and perform the substitutions in the source files. (Right now everything generates is GPL. If that's a problem, [let us know](#).)
 - It ought to let you choose a package name other than `com.sgi.[project]`.
- It should be easy (or at least possible) to replace the privilege subsystem with **sudo** for administrators who've already got it set up & are more comfortable with it. I'm not familiar with sudo myself, but most of the changes would probably be in the `runpriv` command, or in

services/privilege/PrivListener.c++.

- You need to be able to load tasks from multiple products in the TaskManager. Ask John R about this.

Documentation "Bugs"

- The badly-done cascading style sheets should be removed from the tutorials, or rhino.css needs to be included in the tutorials tarball. As it is, it's broken.
- the GUIComponents.html tutorial page could use some work. (See where the "XXX" stuff commented out.)
- Now that we're generating server-side API documentation, we ought to take advantage of the tutorials.
- The tutorials & such should be updated to reflect mkrhinoism's current state.

These web pages

- Move these known problems & missing features into Bugzilla. ("They have computers for like this, you know.")
- Set up the CVS tree. Right now people have to download the whole tarball & send back patches.

[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark](#) in

[products](#)[industries](#)[developers](#)[services & support](#)[open](#)

▼ Rhino

[Overview](#)[News](#)[Documentation](#)[FAQ](#)[Mailing List](#)[How To Contribute](#)[To-Do List](#)[Download](#)[License](#)

The Rhino License

(This is taken from Rhino's COPYING file.)

Rhino is free software; see the copyright notice in each file for the terms under which it may be redistributed and/or modified.

As Rhino is composed mostly of header files & libraries, most files are licensed under the terms of LGPL; in general the standalone commands and test programs are licensed under the terms of the GPL, except for the Java test programs, which are LGPL.

▼ SGI Open Source

[Project List](#)

▼ SGI Support and Resources

[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)

▼ Community Resources

[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark in](#)

[products](#)[industries](#)[developers](#)[services & support](#)[5](#)

open

▼ SGI Open Source[Project List](#)**▼ SGI Support and Resources**[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)**▼ Community Resources**[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)

SGI Open Source Project List

The following projects have either originated within SGI, have SGI employees coordinating the development and maintaining the master trees, or have SGI employees as significant core contrib

Linux® Kernel Work

- [SGI ProPack™ for Linux](#) (contains kernel work and other packages)
- [CpuMemSets](#) (Processor and Memory Placement)
- [KDB](#) (Linux kernel debugger)
- [Kernprof](#) (Kernel Profiling)
- [Lockmeter](#) (Linux kernel lock-metering)
- [NUMA](#) (NUMA support in Linux)

Linux Resource Management Work

- [PAGG](#) (Process Aggregates)

Filesystem & Storage Work

- [Linux FailSafe™](#) (SGI FailSafe for Linux)
- [XFS™](#) (High Performance Journaling File System)
- [fam & imon](#) (File Alteration Monitor and Inode Monitor)

Graphics Projects

- [OpenGL Performer](#) (High-Performance 3D Rendering Toolkit)
- [GLX](#) (OpenGL extensions to X)
- [OpenGL® Sample Implementation](#) (Standard Cross-platform 3D and 2D Graphics API)
- [Open Inventor™](#) (object-oriented toolkit for interactive 3D graphics)

Other Projects

- [PCP](#) (System Performance Monitoring and Management Framework)
- [LKCD](#) (Linux Kernel Crash Dumps)
- [LTP](#) (Linux Test Project)
- [Rhino](#) (Infrastructure for System Administration Applications)
- [Mozilla](#) (also see [SGI Freeware](#))

SGI Freeware Projects and Cool Downloads

- [SGI® Histx 1.1](#) (application performance analysis tools for SGI® Altix® platforms)

[Jump to SGI IRIX Freeware Pages](#) ↻

[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark in](#)

[products](#)[industries](#)[developers](#)[services & support](#)[s](#)[open](#)

▼ **SGI Open Source**

[Project List](#)

▼ **SGI Support and Resources**

[SGI and Linux](#)[Supportfolio for Linux](#)[Customer Services](#)

▼ **Community Resources**

[Newsgroups & Mailing Lists](#)[Support Links](#)[Open Source Sites](#)[Linux Documentation Project](#)

Open Source Sites

The following are sites which provide access to other Linux and Open Source projects relevant to

- [Samba Web pages](#)
- [Red Hat](#)
- [Apache Software Foundation](#)
- [Source Forge](#)

[terms of use](#) | [privacy policy](#) | [questions/c](#)

Copyright © 1993-2003 Silicon Graphics, Inc. All rights reserved. | [trademark in](#)

[products](#)[industries](#)[developers](#)[services & support](#)

5

developer

[Developer Central](#)[SGI Global Developer Program](#)[Join Program](#)[Benefits](#)[FAQ](#)[Forgot your Password?](#)[Member Center](#)[Login](#)[Contact Us](#)[Related Sites](#)[Third Party Applications Dir](#)

SGI Global Developer Program

The Power of Partnership

The SGI Global Developer Program is a program with geographic flexibility that enhance developers' knowledge of the SGI platforms and SGI's business relationships with our developer community. There is **no membership fee** to join the program!

We invite all developers to join our program as a **Developer Online** member. Take advantage of our web-based service that provides vast information on development too techniques, on-line resources as well as some basic business development ideas.

For developers looking to additionally develop joint business opportunities with SGI, we recommend you look at the **Developer Plus** memberships. We provide members with significant development benefits and personal touch to support joint business development activities. Developers wishing to join at either Developer Plus level must develop applications for general distribution and commercial resale and fulfill certain business requirements that are outlined in the Developer Plus legal agreement.

1. **Developer Plus** - for those developing products on IRIX, and possibly Linux. Members receive technical and business development support related to the development and sale their products including:
 - o Free IRIX Developer Tools and Compiler Bundle: list price \$20,000
 - o Significant development system discounts: 50% -70% off local country list
 - o Intel Linux compiler discounts: when purchased with a development system
 - o Competitive support discounts: up to 50% off local country list price
 - o Business development-related activities
2. **Developer Plus for Linux** - For those developers specifically interested in Linux. Members receive the same benefits as Developer Plus, except for discounts specific to IRIX and the free IRIX Developer Tools and Compiler Bundle.

To join the SGI Global Developer Program, developers submit a completed membership form and signed terms and conditions.

Key membership web pages are:

[Membership form](#)[Program Benefits](#)[Hardware and Software Discounts](#)

Developer Membership Support Center

Phone: 800-770-3033 (US & Canada) or 650-933-3033 (worldwide)

Fax: 651-683-5288

[E-mail](#)

Using this site means you accept its [terms of use](#) | [privacy policy](#) | [trademark](#) |
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

[products](#)[industries](#)[developers](#)[services & support](#)[s](#)

developer

◆ [Developer Central](#)▼ [Business Development](#)[Events](#)[Discounts](#)[Partner Profile Template](#)[Contact Us](#)▼ [Related Sites](#)[Member Center](#)[Third Party Applications Dir](#)

Business Development

The Power of Partnership

SGI is committed to working with our developers to create joint revenue opportunities. In this section you will find many useful resources to promote your business and strengthen your relationship with SGI.

Getting Started

- **Advertise your products in the Third Party Applications Directory**
Increase your revenue opportunities by expanding your prospective customer base. Add your products in the [SGI Third Party Applications Directory](#).
- **Leverage the SGI Brand**
You are encouraged to take advantage of our brand recognition by listing the SGI on your web site in a high traffic area with text and a link back to the SGI web site the logo and usage guidelines at http://www.sgi.com/company_info/media_request.html
- **Issue a press release announcing your product's availability on SGI**
SGI supports your efforts in broadening awareness for your products on the SGI platform. [Contact us](#) to notify us about your press release or to request assistance in the development of the press release that may include a quote from SGI for your press release. Note all SGI quotes must be approved by SGI prior to release of your announcement.
- **Create a confidential Partner Profile**
Your Partner Profile provides SGI personnel with information about your company including overview, positioning and strategy. SGI marketing executives, global account managers and SEs use your profile to determine how your products can complement SGI's product line and increase both companies' sales. Create your Partner Profile using this [template](#) and then [email](#) it to us.

Ongoing Activities

- **Participate in SGI sponsored events**
Throughout the year, SGI sponsors and invites Developer Plus level members to participate in a wide variety of events from internal sales meetings to trade show. You are encouraged to work with us to identify potential opportunities that you will be invited to. Check our [Events page](#) and then [contact us](#) to inform us of the appropriate events for your company to participate in.
- **SGI participation in your events**
SGI, on occasion, may participate in events with Developer Plus level members. Participation may include: speakers, equipment loan, press and analyst relations support or marketing collateral distribution. [Contact us](#) to discuss event opportunities.

- **Deliver Product Demos**

Developers have the opportunity to provide SGI with demo copies of their products to be used at the SGI Customer Briefing Center and in the SGI demo centers. These facilities are used for customer presentations, customer benchmarks, etc. and can greatly increase the visibility of your product on the SGI platform.

- **Participate in Partner Needs Studies**

Your input helps us enhance the developer program. We may occasionally request that you participate in various developer studies.

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark info](#)
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

[products](#)[industries](#)[developers](#)[services & support](#)[developer](#)[◆ Developer Central](#)[▼ Developing on SGI](#)[IRIX](#)[Linux](#)[Global Shared Memory](#)[Developer Central](#)[Glossary](#)[Development](#)[Questions !\[\]\(6f6bc8383724387816b6be39e2809aae_img.jpg\)](#)[▼ Related Sites](#)[Join Developer Program](#)[SGI Software](#)[Technical Resources](#)

Developing on SGI

SGI provides scalable compute and graphics systems that can solve today's largest technical and creative problems.

SGI® systems are based on either the MIPS® CPU and IRIX® operating system or the Intel® Itanium® 2 CPU and the Linux® operating system. These systems share a common architecture that provides flexibility for our customers.

Whether your development focus is on high performance computing (HPC), visualization storage, SGI systems provide the following features that enable your applications to scale beyond traditional compute limits:

64-bit computing: expands the size of the problems that can be solved by providing the ability to handle larger scientific simulations, larger data cache, hold greater amount of data in memory.

Global shared memory: eliminates data transfer overhead by providing a single memory address, allowing all processors to access all the data in the system's memory directly and efficiently.

NUMAflex™ architecture: provides balanced resources even for very large systems as CPUs and memory are added, enabling systems to efficiently scale well beyond the traditional bus-based architecture systems

Scalable graphics: allows an application to add graphics resources (multipipes) to meet the requirements of larger visualization problems

SGI Global Developer Program

The [SGI Global Developer Program](#) is the primary source for technical and business resources for anyone developing on the SGI platforms. There is no membership fee to join the program. Our program provides significant benefits to developers including:

- a Web site filled with technical information
- [development system access](#) via the Internet
- [technical assistance](#) for software development and tuning

Contact us today about joining

Developer Membership Center
Phone: 650-933-3033 (worldwide)
Fax: 650-933-0284
[Email us](#)
www.sgi.com/developers/

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark i](#)
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

[products](#)[industries](#)[developers](#)[services & support](#)[developer](#)[Developer Central](#)[Developing on SGI](#)[IRIX](#)[Linux](#)[Global Shared Memory](#)[Developer Central](#)[Glossary](#)[Development](#)[Questions](#)[Related Sites](#)[Join Developer Program](#)[SGI Software](#)[Technical Resources](#)

Developing on SGI

SGI provides scalable compute and graphics systems that can solve today's largest technical and creative problems.

SGI® systems are based on either the MIPS® CPU and IRIX® operating system or the Intel® Itanium® 2 CPU and the Linux® operating system. These systems share a common architecture that provides flexibility for our customers.

Whether your development focus is on high performance computing (HPC), visualization storage, SGI systems provide the following features that enable your applications to scale beyond traditional compute limits:

64-bit computing: expands the size of the problems that can be solved by providing the ability to handle larger scientific simulations, larger data cache, hold greater amount of data in memory.

Global shared memory: eliminates data transfer overhead by providing a single memory address, allowing all processors to access all the data in the system's memory directly and efficiently.

NUMAflex™ architecture: provides balanced resources even for very large systems as CPUs and memory are added, enabling systems to efficiently scale well beyond the traditional bus-based architecture systems

Scalable graphics: allows an application to add graphics resources (multipipes) to meet the requirements of larger visualization problems

SGI Global Developer Program

The [SGI Global Developer Program](#) is the primary source for technical and business resources for anyone developing on the SGI platforms. There is no membership fee to join the program. Our program provides significant benefits to developers including:

- a Web site filled with technical information
- [development system access](#) via the Internet
- [technical assistance](#) for software development and tuning

Contact us today about joining

Developer Membership Center
Phone: 650-933-3033 (worldwide)
Fax: 650-933-0284
[Email us](#)
www.sgi.com/developers/

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark](#) |
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)



United States

products

industries

developers

services & support

5

developer

◆ [Developer Central](#)▼ [Developing on SGI](#)[IRIX](#)

- [IRIX Development Tools](#)
- [Localized Applications](#)
- [Partitioned Environment](#)

[Linux](#)[Global Shared Memory](#)[Developer Central Glossary](#)[Development](#)[Questions](#) ♥ [Related Sites](#)[Join Developer Program](#)[SGI Software](#)[Software Download](#)[Software Keys](#)[Technical Resources](#)

IRIX®

The IRIX® operating system is the leading technical high-performance 64-bit operating system based on industry-standard UNIX®.

IRIX® 6.5 offers system scalability and binary compatibility across the MIPS® processor based family of platforms and is compliant with UNIX® System V Release 4 and The O Group's many standards, including UNIX 95 and POSIX®.

The SGI Global Developer Program supports IRIX developers by providing access to development systems and technical information, as well as development and tuning assistance. Developer Plus members also receive a free IRIX compiler bundle.

IRIX Resources

Hardware

- [Development System Leasing Program](#)
- [Internet Access to Development Systems](#)
- [System Discounts](#)

Software

- [IRIX 6.5](#)
- [IRIX Development Tools and Compiler Bundle](#)
- [Development Tools](#)
- [Software Licensing](#) - how to obtain software licences or upgrades
- [Third Party Application Directory](#)

Support

- [Developer Online Consulting](#) - development and tuning tec assistance
- [Online Technical Resources](#)
- [Support Discounts](#)

Technical Information

- [Application Compatibility in SGI IRIX 6.5](#) - describes the application compatibility protections offered by IRIX 6.5
- [News and Information](#)
- [SGI Technical Publications Online](#)

Using this site means you accept its [terms of use](#) | [privacy policy](#) | [trademark](#) |
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

[products](#)[industries](#)[developers](#)[services & support](#)[developer](#)[Developer Central](#)[SGI Open Source](#)[Newsgroup and Mailing Lists](#)[Support and References](#)[Open Source Sites](#)[Related Sites](#)[Project List](#)[Supportfolio](#)[Linux Documentation Project](#)

SGI Open Source

SGI established its intellectual alliance with Open Source years before the term was coined. That alliance continues to this day through contributions of SGI intellectual energy and properties to the Open Source developmental maelstrom. SGI believes that connecting ever more of the distributed intellectual resources of the Net helps individual developers and the global Open Source community develop alliances and accelerate the growth and maturation of market-like computation ecologies. Thus, this site is another SGI contribution.



This is the Developer Central portal to all SGI Open Source material. Wanderers, seekers and developers are offered this gateway to Open Source technical content and advice, libraries of downloadable software, contribution points, email lists, and links to the Open Source community.

SGI Open Source Projects

Here is where you gain access to the [Open Source projects](#) SGI has released. Each project name provides access to source code download sites, mailing lists, licenses, recent project news, information on how to contribute, and FAQ's.

SGI Support and Resources

SGI is committed to providing enterprise-caliber support and service to Linux and Open Source developers using SGI systems. This section contains links to key SGI support resources, as well as the latest SGI Open Source news, and information on Linux-based products.

Community Resources

There are many Open Source community resources and services online. Referenced here are a few useful and well known newsgroups, support references, and other Open Source and Linux-related information and download sites you may find valuable.

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark](#) |
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

[products](#)[industries](#)[developers](#)[services & support](#)[developer](#)[Developer Central](#)[SGI Open Source](#)[Newsgroup and Mailing Lists](#)[Support and References](#)[Open Source Sites](#)[Related Sites](#)[Project List](#)[Supportfolio](#)[Linux Documentation Project](#)

SGI Open Source

SGI established its intellectual alliance with Open Source years before the term was coined. That alliance continues to this day through contributions of SGI intellectual energy and properties to the Open Source developmental maelstrom. SGI believes that connecting ever more of the distributed intellectual resources of the Net helps individual developers and the global Open Source community develop alliances and accelerate the growth and maturation of market-like computation ecologies. Thus, this site is another SGI contribution.



This is the Developer Central portal to all SGI Open Source material. Wanderers, seekers and developers are offered this gateway to Open Source technical content and advice, libraries of downloadable software, contribution points, email lists, and links to the Open Source community.

SGI Open Source Projects

Here is where you gain access to the [Open Source projects](#) SGI has released. Each project name provides access to source code download sites, mailing lists, licenses, recent project news, information on how to contribute, and FAQ's.

SGI Support and Resources

SGI is committed to providing enterprise-caliber support and service to Linux and Open Source developers using SGI systems. This section contains links to key SGI support resources, as well as the latest SGI Open Source news, and information on Linux-based products.

Community Resources

There are many Open Source community resources and services online. Referenced here are a few useful and well known newsgroups, support references, and other Open Source and Linux-related information and download sites you may find valuable.

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark](#) |
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

▼ **Trademark Information**

[Proper Use](#)

[Guidelines](#)

[More Information](#)

[Questions](#)

▼ **Trademark Lists**

[SGI](#)

[Other Companies](#)


[Terms of Use](#)

SGI Trademark List

The following are selected trademarks and service marks (as indicated) of Silicon Graph Inc., except as otherwise indicated by hyperlinked text.

An "®" notice indicates that the mark is registered in the United States.

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [J](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [SGI](#) [Silicon Graphics](#) [T](#) [U](#) [V](#) [W](#) [X](#)
[numeric](#)

Trademark	Descriptor
4D Ada™	software
4DDN™	networking option
4DLT™	networking option
AstroCompress™	board
Bandwidth Engine™	<i>marketing slogan used in connection with Origin® servers</i>
CASEVision™	software, C++ development environment
CASEVision™/ClearCase	software configuration management system
CASEVision™/Tracker	defect tracking software
CASEVision™/WorkShop	programming environment, software
Cellular IRIX™	operating system software
Challenge®	server
 ™	server
Challenge® 10000	server
Challenge® DM	network resource server
Challenge® DM R4400® <i>*see note re-attribution*</i>	network resource server
Challenge® L	network resource server
Challenge® S	network resource server
Challenge® S R5000® <i>*see note re-attribution*</i>	network resource server
Challenge® XL	network resource server
ChallengeArray™	server
ClearCoat™	reflection map feature
ClearCoat™ 360	reflection map feature
Cobalt™	graphics chipset
Coloratura™	color adjustment software

ColorLock™
Crimson™
Crimson Elan™
CXFS™
Developer Magic™



[the Developer Magic logo]™

DMediaPro™
Elan Graphics™
FailSafe®

FDDI Visualizer™
FDDIXPress™
Floops™



[Floops]™

Geometry Accelerator™
Galileo Video™
Geometry Engine®
Geometry Partner™
GIGAchannel™
GL™
GLX™
Graphics Library™
ignite your mind™
ImageVision™
ImageVision Library®
Impressario™
Indigo®
Indigo Elan™
Indigo Magic™
Indigo Video™
Indigo2™
Indigo2 High Impact™
Indigo2 Impact™
Indigo2 Impact™10000
Indigo2 Maximum Impact™
Indigo2 Solid Impact™
Indigo2 Video™
Indy® see note re attribution

IndyCam™
see note re-attribution

color calibration feature
workstation
workstation
clustered version of XFS filesystem
software development environment
software development environment

digital media product
graphics subsystem
family of system software for high-availability clustering (s
IRIS FailSafe, Linux FailSafe, SGI FailSafe)
network management software
network interface controller board and software
VRML animated character
VRML animated character

processor
video software
processor
program
(see SGI® Origin® 200 GIGAchannel™)
API (application programming interface)
API (application programming interface)
API (application programming interface)
(corporate slogan)
C++ image processing toolkit
toolkit
printing and scanning software
workstation, system
workstation, system
desktop environment, graphical user interface
video option card
workstation, system
workstation, system
workstation, system
workstation, system
workstation, system
flat panel display
workstation, system
digital color video camera

Indy Modeler™
see note re-attribution

Indy Presenter™
see note re-attribution

Indy® R5000®
see note re-attribution

Indy Studio™
see note re-attribution

Indy Video™
see note re-attribution

InfinitePerformance™

InfiniteReality® (no space between the words please!)

InfiniteReality2™

InfiniteReality3™

InfiniteReality4™

InPerson®



[the InPerson logo]™

Integrated Visual Computing™

Interactive Shading Language™

Inventor™

IRGO™

IRIS®

IRIS 4D™

IRIS® 5080 Emulator

IRIS Annotator™

IRISconsole™

IRIScope™

IRIS Crimson™

IRIS Crimson Elan™

IRIS Explorer™

IRIS FailSafe®

IRIS Geometry Pipeline™

IRIS GL™

IRIS Gold Seal™

IRIS® Graphics Card

IRIS Graphics Library™

IRIS Impressario™



[the IRIS Impressario logo]™

IRIS Indigo™

IRIS Indigo Elan™

IRIS InSight™

IRIS Inventor™

workstation, system

flat panel display

workstation, system

workstation, system

flat panel display

graphics subsystem

graphics subsystem

InfiniteReality graphics subsystem enhancement

InfiniteReality graphics subsystem enhancement

graphics subsystem

conferencing software

conferencing software

Architecture

technology

object-oriented developer's toolkit

HPC workflow-optimization features

hardware, software

system

networking product

multimedia software application

multiserver management system

N/A

workstation, system

workstation, system

user environment, application creation system

system software

software

programming toolkit

compatibility certification

(graphics card)

programming toolkit

printing and scanning environment

printing and scanning environment

workstation, system

workstation, system

online viewer

3D graphics programming toolkit

IRIS Partner™	program
IRIS Performer™	software development environment
IRIS POWER C™	software option
IRIS POWER Series™	workstation, shared-memory multiprocessing system
IRIS POWER STATION™	workstation, system
IRIServer™	software
IRIShare™	networking solution
IRIS Showcase™	multimedia authoring and presentation software
IRIS Universe™	magazine
IRISview™	software application
IRIS ViewKit™	class library, application framework
IRISVISION™	high-resolution graphics boards, DOS software developer
IRIS WorkSpace™	user environment
IRIX®	operating system software
IRIXCare™	support program
IRIX Networker™	data management tool, storage management software
IRIXPro™	tools
IRIXview™	software
ISL™	technology
IVC™	Architecture
Key-O-Matic™	automatic license fulfillment system
Lavarand™	random number generation system
Linux FailSafe™ (see special attribution)	system software
Maya® (see special attribution for Alias Wavefront)	software
Media Commerce™	initiative or solution
MindShare™	software, collaborative environment
MineSet™	data mining and visualization tools
miPS® (see special attribution)	compiler
NetVisualyzer™	networking software
NUMAflex™	feature, benefit
NUMAlink™	interconnect technology, cables
NUMAlink™ 3	interconnect technology, cables
NUMAlink™ 4	interconnect technology, cables
(for O2, see Silicon Graphics® O2®)	
O2Cam™	digital video camera
O2 Studio™	workstation, system
(for Octane, see Silicon Graphics® Octane®)	
(for Octane2, see Silicon Graphics® Octane2™)	
OnAir™	graphics software
Onyx4™ (see Silicon Graphics® Onyx4™)	visualization system
Onyx4™ UltimateVision™ (see Silicon Graphics® Onyx4™ UltimateVision™)	visualization system

(for Onyx workstation, see Silicon Graphics® Onyx®)

workstation

(for Onyx visualization systems, see SGI® Onyx®)


visualization systems

(for Onyx2 workstation, see Silicon Graphics® Onyx2®)


workstation

OpenGL®

API (application programming interface)

 [the original OpenGL logo]™

API (application programming interface)

 [the current OpenGL logo]®

API (application programming interface)

OpenGL® ES

API (application programming interface)

 [OpenGL ES logo]

API (application programming interface)

OpenGL Multipipe™

graphics system software

 [OpenGL Multipipe logo]™

graphics system software

OpenGL Multipipe™ Software Development Kit (SDK)

graphics software solution

OpenGL on a Chip™

initiative

OpenGL Optimizer™

API (application programming interface)



API (application programming interface)

[the OpenGL Optimizer logo]™

OpenGL Performer™

real-time graphics API

OpenGL Shader™

software development kit

OpenGL® Shading Language


API (application programming interface)

OpenGL Vizserver™

computing solution

OpenGL Volumizer™

API (application programming interface)

 [OpenGL Volumizer logo]™

API (application programming interface)

Open Inventor™

3D developer's toolkit

 [the Open Inventor logo]™

3D developer's toolkit

OpenML®

open standard for digital media libraries

 [the OpenML logo]™

open standard for digital media libraries

OpenMP™

multi-platform shared-memory API (application programming interface)

 [the OpenMP logo]™

multi-platform shared-memory API (application programming interface)

OpenVault™

open media manager



server

(for Origin server, see SGI® Origin®)

server

Performance Co-Pilot™

software tools

Performer™	software development environment
Personal IRIS™	workstation, system
Power Center™	server
Power Challenge™	server
Power Challenge™ 10000	server
Power ChallengeArray™	software environment
Power Channel™	server
powered by sgi [the Powered by SGI logo]	corporate slogan
Power Fortran Accelerator™	software
Power Indigo2™	workstation, system
Power Indigo2 Impact™	workstation, system
Power Indigo2 Maximum Impact™	workstation, system
Power IRIS™	workstation, system
Power Onyx™	graphics supercomputer
PowerPath™	coherent interconnect, architecture
PowerPath-2™	coherent interconnect
Power Series™	workstation
Power Store™	removable module for the Power Center electrical power distributor
PowerVision™	graphics supercomputer
ProDev™	software
ProDev™ WorkShop	software
Professional IRIS™	workstation, system
R10000® (<u>see special attribution</u>)	microprocessor
R12000® (<u>see special attribution</u>)	microprocessor
R12000A™ (<u>see special attribution</u>)	microprocessor
R14000™ (<u>see special attribution</u>)	microprocessor
R14000A™ (<u>see special attribution</u>)	microprocessor
R16000™ (<u>see special attribution</u>)	microprocessor
RapidApp™	software
REACT™	software
REACT/pro™	software
Reality Center® (<u>see SGI® Reality Center®</u>)	visualization facility; collaborative virtual reality technology
Reality Center® 1000D	visualization desk
Reality Center® 2000D	visualization desk
Reality Center® 3300W	visualization wall display
Reality Center Insight™	high-resolution display
RealityEngine™	graphics subsystem
RealityEngine2™	graphics subsystem
RealityMapping™	3D software
RealityMonster®	system
Reality Station™	system

RoboInst™

S2MP™
(referred to as ccNUMA)

see what's possible™

SGI®



SGI® 1100

SGI® 1200

SGI® 1400

SGI® 1400L

SGI® 1400M

SGI® 1450

SGI® Advanced Cluster Environment (ACE)

SGI Advanced Linux™ Environment
(see special attribution)

SGI® Altix® 350

SGI® Altix® 3000

SGI® Altix® 3300

SGI® Altix® 3700

SGI® CoOpCare™

SGIconsole™

SGI DataSync™

SGI® Decision Support Center

SGI FailSafe™

SGI Federal™



[government subsidiary logo]

SGI® File Server

SGI® FullCare™

SGI® FullExpress™

SGI® FullProduction™

SGI GenomeCluster™

SGI® Global Developer Program

SGI® Global Services

SGI Graphics Cluster™

SGI HPC ON-DEMAND™

SGI ImageSync™

SGI® InfiniteStorage

SGI® Insider

SGI Internet Server™

SGI Linux™ (see special attribution)

SGI Media Server™

software

architecture

[corporate slogan]

corporate brand name; technology, servers

corporate brand name; technology, servers

server

server

server

server

server

server

advanced cluster environment

open-source operating system; software

server

family of servers and superclusters, family, server(s),
supercluster(s)

server

supercluster

customer support service offerings

remote multiserver management system

data synchronization technology

solution

family of system software for high-availability clustering (s
IRIS FailSafe, Linux FailSafe, FailSafe)

government subsidiary brand name

government subsidiary brand name

system

customer support service offerings

customer support service offerings

customer support service offerings

bioinformatics solution

education, training, and business development program

services

high-performance visualization cluster

services

precision channel synchronization technology

storage solution; hardware and software

online customized news digest

server; system

open source operating system

server

SGImeeting™
 SGI® Mobile Innovation Center
 SGI® NUMA
 SGI - One step ahead™ (see special note)
 SGI® Onyx®
 SGI® Onyx® 300
 SGI® Onyx® 3000 series
 SGI® Onyx® 3200
 SGI® Onyx® 3400
 SGI® Onyx® 3800
 SGI® Origin®
 SGI® Origin® 200
 SGI® Origin® 200 GIGACHannel™
 SGI® Origin® 2000 series
 SGI® Origin® 2000
 SGI® Origin® 2100
 SGI® Origin® 2200
 SGI® Origin® 2400
 SGI® Origin® 2800
 SGI® Origin® 300
 SGI® Origin® 3000 series
 SGI® Origin® 3200
 SGI® Origin® 3200C
 SGI® Origin® 3400
 SGI® Origin® 3800
 SGI® Origin® 3900
 SGI Pro64™
 SGI ProPack™
 SGI ProPack™ for Linux® (see special attribution)
 SGI® Reality Center®
 SGI® Reality Center® 3300W
 SGI® Reality Center Insight™
 SGI® Reality Centre™
 (British spelling—see special note)
 SGI SAN Server™ 1000
 SGI® Service in a Box
 SGI® ShotMaker
 SGI® SpeedShop™
 SGI® StudioCentral™ Library
 SGI® Total Performance 900
 SGI® Total Performance 9100
 SGI® Total Performance 9300
 SGI® Total Performance 9400

conference tool software
 demonstration facility
 architecture
 corporate slogan
 visualization system family
 visualization system
 visualization system series
 visualization system
 visualization system
 visualization system
 server family
 server
 server
 server series
 server
 server
 server
 server
 server
 server
 server series
 server
 clustered server
 server
 server
 server
 compiler; technology
 software feature for Linux OS
 open source feature for Linux OS

 visualization facility; collaborative virtual reality technology
 visualization facility; collaborative virtual reality technology
 display
 visualization facility; collaborative virtual reality technology

 system
 customer service contract package
 real-time audio and video suite of software
 integrated software package of performance tools
 system
 SCSI storage system
 Fibre Channel storage system
 Fibre Channel storage system
 Fibre Channel storage system

SGI® Total Performance 9500

SGI® TP900

SGI® TP9100

SGI® TP9300

SGI® TP9400

SGI® TP9500

SGI® Visual Serving™

SGI® Visual Serving Initiative™

SGI Zx10™

SHMEM™

Showcase™

Silicon Campus™

Silicon Exchange™



[the original Silicon Exchange logo]™



[the current Silicon Exchange logo]™

Silicon Graphics®



® [the SGI cube]

Silicon Graphics® 230

Silicon Graphics® 320

Silicon Graphics® 330

Silicon Graphics® 540

Silicon Graphics® 550

Silicon Graphics® 750

Silicon Graphics® 1600SW

Silicon Graphics® DS1100

Silicon Graphics® F180

Silicon Graphics® F181

Silicon Graphics® F220

Silicon Graphics® C220

Silicon Graphics Fuel®

storage array

SCSI storage system

Fibre Channel storage system

Fibre Channel storage system

Fibre Channel storage system

storage array

end-to-end solution using OpenGL Vizserver™ technology

collaborative effort program

server, system

API (application programming interface), library

authoring and presentation software

products, sales program

integrated software solution set

integrated software solution set

integrated software solution set

technology, workstations, systems, software, peripherals (applicable), customer support services, training services, w sites, resource centers

technology, workstations, systems, software, peripherals (applicable)

technology, workstations, systems, software, peripherals (applicable)

technology, workstations, systems, software, peripherals (applicable)

technology, workstations, systems, software, peripherals (applicable)

technology, workstations, systems, software, peripherals (applicable)

workstation

workstation desktop model

workstation

workstation deskside tower model

workstation

workstation; system

flat panel monitor

disk array

flat panel display

flat panel display

flat panel display

flat panel display

visual workstation

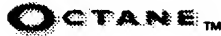
Silicon Graphics MultiLink™

Silicon Graphics® O2®



Silicon Graphics® O2+™

Silicon Graphics® Octane®



Silicon Graphics® Octane2™

Silicon Graphics® Octane2™ V6

Silicon Graphics® Octane2™ V8

Silicon Graphics® Octane® CADduo

Silicon Graphics® Octane®/MXI

Silicon Graphics® Octane®/SI

Silicon Graphics® Octane®/SSI

Silicon Graphics® Onyx®

Silicon Graphics® Onyx® 10000

Silicon Graphics® Onyx® InfiniteReality®



[the InfiniteReality logo]™

Silicon Graphics® Onyx® i-Station™

Silicon Graphics® Onyx2®



Silicon Graphics® Onyx2® InfiniteReality®

Silicon Graphics® Onyx2® InfiniteReality2™

Silicon Graphics® Onyx2 Reality™

Silicon Graphics® Onyx2® RealityMonster®

Silicon Graphics® Onyx4™

Silicon Graphics® Onyx4™ UltimateVision™

Silicon Graphics® Presenter 1280

Silicon Graphics® SD1100

Silicon Graphics® Tezro™

Silicon Graphics Zx10™

Silicon Graphics Zx10™ VE

Silicon Junction™



Silicon Mine™

Silicon PRESS™

Silicon Service™

Silicon Stage™

Silicon Studio®



[the Silicon Studio logo]

adapter

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

workstation, system

visualization system

visualization system

flat panel display

serial digital interface board

visual workstation

workstation, system

workstation, system

corporate intranet

corporate intranet

data mining and analysis tools

software

support services

software

multimedia application software, training facilities, web site

educational, training and information services and facilities
applicable)

TM



[the Silicon Studio logo]™

Silicon Studio/LA™

Silicon Studio/London™

Silicon StudioLive™

Silicon Surf™

SiliconWorks™



[the SiliconWorks logo]™

Sirius Video™

SkyWriter™

Stereo View™

StreamCaster™

Streaming Multiport Architecture™

StudioLive™

StudioServices™

StudioSupport™

StudioTraining™

SuperWide™

Supportfolio™

SynaptIQ™

Tezro™ (see Silicon Graphics® Tezro™)

The New Force in Computing™

The solution is in sight™

The Source of Innovation and Discovery™

Trusted IRIX™

UPSafe®

Virtual 24™

VPro™

Wahoo Technology™

WebFORCE® Intranet Junction

WebMagic™ Author

WebMagic™ Pro

WorldView™

XFS®

XIO™

educational, training and information services and facilities applicable)

educational and training services and facilities

educational and training services and facilities

online tools and information services

corporate web site

manufacturing industry solutions

manufacturing industry solutions

video option

visual system

goggles

server software

tuned system architecture for performance improvement

online services

tools

system support services

educational services

monitor format

software tools, customer support service offerings

software toolkit

visual workstation

(corporate slogan)

(corporate slogan)

(corporate slogan)

system software

bundled hardware, software, and service package

color graphics

graphics subsystem

tuned system architecture for performance improvement

Web-based client server software application

software

software

native language support products

filesystem software

hardware; graphics board

Trademark Notices

® indicates a trademark registered *in the United States*; which is either a trademark or a registered trademark in all other countries where it is used.

™ indicates a trademark that is not registered in the United States.

SM indicates a service mark that is not registered in the United States.

Trademark Attribution

Please include the following trademark attribution in any written marketing material that reference trademarks:

[mark as it appears above] is a trademark *[or service mark or registered trademark, as indicated]* Silicon Graphics, Inc., in the United States and/or other countries worldwide.

Trademark attributions for particular marks above:

Challenge® DM R4400®

Challenge is a registered trademark of Silicon Graphics, Inc. R4400 is a registered trademark of Technologies, Inc. used under license by Silicon Graphics, Inc.

Challenge® S R5000®

Challenge is a registered trademark of Silicon Graphics, Inc. R5000 is a registered trademark of Technologies, Inc. used under license by Silicon Graphics, Inc.

Indy®

Indy [and any other Indy trademark above] is/are (a) registered trademark(s) used under license United States and owned by Silicon Graphics, Inc., in other countries worldwide.

Linux FailSafe™

FailSafe is a registered trademark and Linux FailSafe is a trademark of Silicon Graphics, Inc., in United States and/or other countries worldwide. Linux is a registered trademark of Linus Torvalds several countries, used with permission by Silicon Graphics, Inc.

MIPS® R10000® and/or R12000® and/or R14000™ and/or R14000A™ and/or MIPSpro™ United States Attribution

MIPS, R10000, and R12000 are registered trademarks and R14000, R14000A, and MIPSpro are trademarks of MIPS Technologies, Inc., used under license by Silicon Graphics, Inc., in the United States and/or other countries worldwide.

R10000®, R12000®, R14000™, R14000A™, and MIPSpro™ United States Attribution (with MIPS brand name)

R10000 and R12000 are registered trademarks and R14000, R14000A, and MIPSpro are trademarks used under license by Silicon Graphics, Inc., in the United States and/or other countries worldwide.

SGI Linux™ and/or SGI Advanced Linux™

SGI Linux™ and SGI Advanced Linux™ are trademarks of Silicon Graphics, Inc. Linux is a registered trademark of Linus Torvalds, used with permission by Silicon Graphics, Inc.

SGI ProPack™ for Linux®

SGI ProPack is a trademark of Silicon Graphics, Inc. Linux is a registered trademark of Linus Torvalds, used with permission by Silicon Graphics, Inc.

Special Notes

SGI - One step ahead™

The "One step ahead" slogan may only be used as a lockup to the SGI logo. Use the One step ahead lockup created by official sources; do not create your own. Do not use "One step ahead" as text or a headline. When using the locked up One step ahead logo, the trademark attribution should read:

"SGI, the SGI logo, and One step ahead are trademarks of Silicon Graphics, Inc."

SGI® Reality Centre™

To be used only in countries that use British spelling, such as Australia, New Zealand, Canada, t
United Kingdom, and certain Asia-Pacific regions.

*The above list is not intended to constitute an exhaustive list of trademarks, slogans, goods
descriptions or other intellectual property rights of SGI, and any omission of same does not cons
a waiver of any kind of SGI's intellectual property.*

[Using this site means you accept its terms of use](#) | [privacy policy](#) | [trademark i](#)
Copyright © 1993-2004 Silicon Graphics, Inc. All rights reserved. | [contact us](#)

Depending on the product, it may be desirable to write one priv command that provides all of the functionality necessary, or several priv commands that are specialized. For example, a product that manages user accounts could have one `userAccount` priv command that could add, remove, and modify accounts based on the arguments, or the product could have `addUserAccount`, `removeUserAccount`, and `modifyUserAccount` priv commands.

When deciding on how to break the functionality into priv commands, remember that the `runpriv` command gives an administrator the ability to grant privileges to users on a *per priv command* basis. Therefore, if you feel that an administrator would benefit from being able to grant privileges to a subset of the functionality of your product, you should divide the priv commands appropriately. For example, in the `userAccount` case above, if there is only the `userAccount` priv command, then the administrator must grant a user permission to perform all user account functions, or grant no permission. In the case where three priv commands were provided, the administrator could give a particular user permission to add user accounts without giving that user permission to modify or delete existing accounts.

Naming Priv Commands

Because all priv commands reside in a common directory, give priv commands names that will not collide with the priv commands written for other products. It is recommended that you prefix the name of your product's priv commands with a prefix representing the product. For example, `fsmgrAddMachine` would perform the "add machine" functionality for the Failsafe Manager (`fsmgr`) product.

Returning Status from Priv Commands

While the Rhino infrastructure gives programmers direct access to the return codes and output streams of priv commands, there are some conventions that make things easier:

1. The priv command should return 0 if it was successful. Otherwise return a non-zero error code. This error code can be used by the GUI to present a user-friendly, localized error message, so make sure that the error code is specific enough for the GUI to display a useful message.
 2. If the priv command is not successful, then the priv command may send any error output to `stderr`. By default, the GUI will display this text if the return value is not 0.
 3. Any other output that the GUI needs should be sent to `stdout`.
-

Validating Input

GUIs written with the Rhino infrastructure often verify that all the arguments to a priv command are valid. This does **not** mean, however, that the priv command can assume that the arguments are valid. There are several reasons for this:

- The priv command may have been invoked from a shell or script and not from the GUI
- The GUI could have a bug
- The GUI may not be able to validate all inputs
- Due to timing issues, the GUI might not know the correct current state of the system for

validation.

For these reasons, it is the responsibility of the priv command to verify that all the inputs are valid before performing any operation.

Priv Commands Should be Atomic

From the user's point of view, a priv command should be an atomic operation that either succeeds completely or fails without making any changes to the system. This is because a half-completed priv command will often leave the system in an inconsistent state that is difficult to diagnose and fix. While this is not always practical, it should be a goal for any priv command.

On a related note, there is nothing to prevent two GUIs from calling the same priv command at the same time. If the system could be corrupted as a result of two or more simultaneous priv commands running, then it is the responsibility of the writer of the priv command to provide some kind of locking mechanism to prevent corruption.

Priv Commands are Logged

All priv commands executed by runpriv are logged in `/var/sysadm/salog`, and `salog` is world-readable. This has several implications:

1. Make sure that no private or secret data is revealed by the priv command on the command line. A way to pass data to a priv command in a secure fashion is discussed below.
 2. An user can see a list of all the priv commands that were run. This let users create scripts that call the priv commands simply by cutting and pasting from the log file. If the user is not root, then they will have to use the runpriv command to run the priv commands.
 3. The log can often be useful while debugging the GUI and the priv commands to see exactly what commands the GUI ran (or tried to run).
-

Passing Arguments to Priv Commands

This section describes the parameter passing conventions used by Rhino applications and the priv commands that they use. The conventions described below are guidelines only - the Rhino infrastructure will allow complete control of the arguments used to start a priv command - but following the conventions will make writing both the GUI Tasks and the priv commands easier because you can take advantage of existing infrastructure for passing the arguments from the client to the server.

The following sections have a lot of detail so that the reader will understand the implications of the way that Rhino passes arguments. Most of the details are taken care of by the infrastructure and libraries supplied with Rhino.

Basic Arguments

Arguments should be passed as key/value pairs, where both the key and value are strings, and the key is

separated from the value by a "=" character. The order of the pairs is not important. The following "escapes" are used:

Character Escape

=	%3d
\n	%0a
%	%25

For example, the priv command to add a user might look like:

```
/usr/sysadm/privbin/addUser username=guest uid=123 homedir=/usr/people/guest "realna
```

This is the way that the client side "runPriv" Java method sends the arguments by default. The key/value style of argument matches the structure of TaskData (the data structure that the Tasks use to store the input collected from users). This makes it so that the Task can automatically convert the data entered by users in a Task to a command line. It also makes the log file easier to understand than the traditional "flag" style of argument specification.

Passing Arrays

There are times when a priv command needs to accept an array of arguments. For example, to add a host to the /etc/hosts file, the priv command might take as many "alias" fields as necessary. In this case, the preferred method is:

1. One key specifies the number of values
2. Each value is passed with a separate key, where the key is formed by appending a number to a prefix.

For example:

```
addHost numAliases=3 alias0=bonnie alias1=bonnie.engr alias2=bonnie.engr.sgi.com
```

This approach obviates the selection of a delimiter character (required in the case of passing an array as, for example, Item=value0,value1,value2) and allows a consistent approach across CLIs.

Passing Args on Stdin

There are some situations where there are too many arguments to fit on the command line, or it's not desirable (perhaps for security reasons) to put a particular argument on the command line. In this case, the priv command can take some arguments on the command line, followed by the special argument "-input". This is a signal to the priv command that it should read the remaining arguments from file descriptor 0 (stdin). The arguments specified on file descriptor 0 are specified as key/value pairs similar to command line arguments, but there are a few differences. Each argument sent to stdin follows the following format:

1. An 8 character hexadecimal ASCII representation of the number of bytes taken up by the key, the value, an equal sign, and a newline.
2. A space
3. The key, quoted as described above
4. An equal sign
5. The value, quoted as described above

6. The newline character

The GUI automatically sends any arguments that don't fit on the command line to stdin. It also sends any piece of TaskData that have been marked as hidden (via `setAttrVisible` method of TaskData or AttrBundle) to stdin. The C API (described below) makes reading all of the arguments, both from the command line and stdin, as easy as a few function calls.

C API

To make the writing of priv commands easier, a C API and library are provided that make the reading of arguments a trivial process. The function calls are described first, followed by example code that illustrates how the API is intended to be used.

This is not meant to be a complete description of the SaParam API. See SaParam.h for complete documentation.

Access

Headers for this API are obtained from the header file `sysadm/SaParam.h` (in `sysadm_root.sw.hdr`). The library that implements the API is `/usr/lib32/libsysadmParam.so`. (in `sysadm_root.sw.lib`)

Types

The types defined by this API are opaque:

```
# SaParam is a structure that holds all of the parameters, not a single param
typedef struct _SaParam SaParam;
typedef struct _SaParamIter SaParamIter;
```

Create and Destroy

```
extern SaParam *SaParamCreate(void);
extern void SaParamDestroy(SaParam *param);
```

`SaParamCreate()` returns NULL if `malloc()` fails.

Set and Get

```
extern int SaParamSet(SaParam *param, const char *key, const char *value);
extern const char *SaParamGet(SaParam *param, const char *key);
```

`SaParamSet()` returns 0 if successful, -1 if `malloc` fails.

`SaParamGet()` returns NULL if there is no value for "key".

The pointer returned by `SaParamGet` is owned by "param", and will remain valid as long as `SaParamDestroy()` or `SaParamSet()` for this key are not called.

Argument Parsing

```
#define SaPARAM_INPUT_ARG "--input"
extern int SaParamParseArgs(SaParam *param, int argc, char *argv[]);
```

Parse command line arguments of the form "key=value" into key/value pairs. If the argument `SaPARAM_INPUT_ARG` ("-input") is encountered, read key value pairs from file descriptor 0 as well. Returns 0 if successful, -1 if memory is exhausted, if read fails, or if an unrecognized argument is encountered.

Enumerating the Keys

```
extern SaParamIter *SaParamIterCreate(SaParam *param);
extern void SaParamIterDestroy(SaParamIter *iter);
extern const char *SaParamIterGetKey(SaParamIter *iter);
extern const char *SaParamIterGetValue(SaParamIter *iter);
extern void SaParamIterReset(SaParamIter *iter);
```

To iterate over all of the keys in an `SaParam` object, create an `SaParamIter` using the `SaParamIterCreate` function, and call `SaParamIterGetKey` repeatedly until it returns `NULL`. At any point in the iteration, `SaParamIterGetValue` can be called to get the value corresponding to the last key returned by `SaParamIterGetKey`. This is more efficient than calling `SaParamGet` with each key.

Priv command Sample Code

```
int main(int argc, char *argv[])
{
    const char *name = NULL;
    const char *uid = NULL;

    // Create param object
    SaParam param = SaParamCreate();

    // Parse the command line.
    SaParamParseArgs(param, argc, argv);

    name = SaParamGet(param, "name");
    uid = SaParamGet(param, "uid");

    ...
    SaParamDestroy(param);
}
```

Decoding an array

```
numParamsString = SaParamGet(params, "numParams");
if (numParamsString != NULL) {
    numParams = atoi(numParamsString);
    for (i = 0; i < numParams; i++) {
        sprintf(buf, "param%d", i);
        param[i] = SaParamGet(params, buf);
    }
}
```

Perl API

The following Perl code provides similar functionality as the C API. It is not as complete as the C API, but it is included here for reference. It currently doesn't parse the arguments from stdin, but that functionality can be added if necessary. If this code is useful, it can be added to the Rhino infrastructure.

Priv command Sample Code

```
#!/usr/bin/perl -w

# Parse the command line.
# Pass a reference to @ARGV, and a reference to a hash that will be
# filled in with the arguments
sub parseArgs {

    # Define the "escape" sequences used to quote the command line.
    # These must match those in SaParam.c
    my %escapes = ("3d" => "=", "0a" => "\n", "25" => "%");

    my ($argv, $out) = @_;
    my ($key, $value);
    foreach (@{$argv}) {

        # Split key and value on the "=" character.
        ($key, $value) = split(/=/);

        # Globally replace any sequence of a "%" character
        # followed by two characters with the character from
        # %escapes. If the sequence is not found in %escapes, then
        # don't replace anything.
        $key   =~ s/%(\w\w)/$escapes{$1}||$%/eg;
        $value =~ s/%(\w\w)/$escapes{$1}||$%/eg;
        $out->{$key} = $value;
    }
}

&parseArgs(\@ARGV, \%args);
foreach $key (sort keys %args) {
    print "$key == $args{$key}\n";
}
...
```

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

Related: [Basic Concepts](#) | [Architecture](#) | [Association](#) | [Category Names](#)

Item and Category in Rhino

This document describes how Category(s) and Item(s) are supported in the Rhino infrastructure. The first section discusses the underlying concepts behind Category(s) and Item(s). The second section presents the API for implementing Category(s) and Item(s) at the server side. The last section presents the API to obtain dynamic information about a specific Category and Item at the client side. Typical clients of Category(s) and Item(s) include Rhino UIs, such as [Tasks](#) and [ItemViews](#).

Underlying Concepts

The fundamental data types used in Rhino are the **Attribute** class and the **AttrBundle** class. An Attribute is a typed key-value pair. The different Attribute types supported are boolean, long, double and string. AttrBundle is an aggregation of Attribute(s). Each AttrBundle instance has two string fields representing the *type* and *selector*. Subclasses of AttrBundle interpret these fields in different ways.

An Item is a subclass of AttrBundle. The type of an Item instance is the name (selector) of the Category instance that it belongs to. The selector of an Item instance is the unique name of the instance within the Category instance. For example, a user account Item instance can have selector *foo* within the type *UserAccountCategory* and the following Attributes:

```
string, userName, foo
long,   userId,  3944
```

Arrays of values can be represented in an Item as follows:

1. One attribute specifies the number of values in the array.
2. Each value is accessed by appending a number to a prefix for the series of values.

For example, if an Item of type group account has an array of Strings representing names of users that use its group id, this would be represented by an Attribute with key, for example, NUM_USERS that specifies the number of values in the array and a key, for example, USER as the prefix to use. USER0, USER1 ... USER<NUM_USERS - 1> would be the Attribute keys of the actual String values. For example, NUM_USERS = 3, USER0 = foo, USER1 = bar, USER2 = baz.

This is the format that the [Association](#) mechanism relies on for parameters referring to the selectors of Item(s) to monitor the relationship between Item(s).

A Category class has a collection of Items. Category is also a subclass of AttrBundle. The type of a Category instance is the constant value *Category*. The selector is the unique name of the Category instance within the system. For example, the user account Category instance would have selector *UserAccountCategory*. A Category can have a set of Attributes that apply to all Item(s) of that type. For example, *UserAccountCategory* instance can have Attributes that store information about whether shadow passwords are in use by a system. Category classes provide API to support monitoring of the Item(s) and Attribute(s) of the specific type and notification of current state and state changes to

interested clients.

Clients interested in information about a Category instance do so via a **CategoryListener** interface. A CategoryListener instance can be registered with a Category for notifications. Upon registration, the CategoryListener instance receives information about the current state of the Category instance. If the state of the Category changes the CategoryListener will receive information about the changes as they occur. Information from Category can be obtained at several granularities.

All client-server communication is asynchronous so that the UI can be responsive to user input and not block waiting for completion of a request to the server. Asynchronous nature is achieved by using a callback model.

The concepts of Category, Item and CategoryListener are implemented in Java at the client-side and in C++ at the server-side. An application developer writes the logic to determine the set of Item(s) (of a specific type) and Category Attribute(s) and to monitor the system for any changes to the Item(s) and Category Attribute(s) in C++, using the server-side API. The application-specific clients, such as specific Tasks or ItemViews are written in Java and use the client-side API to obtain information about Category(s) and Item(s).

Implementing Category(s) and Item(s) on the server-side: C++

All application-specific entities are instances of Items. No subclassing is required. A Category class needs to be subclassed. An instance of the subclass performs application-specific operations to obtain the state of the system and to inform the Category base class of any changes to the state. For example, the *UserAccountCategory* instance (of Category) would read and monitor the passwd files or NIS maps to monitor user account Item(s) to obtain the current state and detect changes.

When the first CategoryListener is added to a Category instance, the Category base class calls `Category::startMonitor()`. Category subclasses must override this method to do whatever is necessary to discover existing Item(s) and monitor Item(s) of the specific type. Information about all Item(s) that exist at the time `Category::startMonitor()` is called should be communicated to the Category base class via `Category::addItem()` calls. Information about Category attributes should be communicated by the subclass via `AttrBundle::setAttr()`. The end of the Item(s) and Category Attribute(s) that exist when `Category::startMonitor()` is called should be communicated to the Category base class via an `Category::endExists()` call. Any future addition, removal of Item(s) as well as changes to the Item(s) should be communicated to the Category base class via `Category::addItem()`, `Category::removeItem()` and `Category::changeItem()` calls. Information about changes to Category attributes should be communicated by the subclass via `AttrBundle::setAttr()`.

Category also supports methods `Category::beginBlockChanges()` and `Category::endBlockChanges()` that can be called by subclasses to indicate the start and end of a block of notifications. `Category::replaceItemList()` can be called by subclasses when it wants to replace the current list of Item(s) by a new list. The Category base class computes any changes between its previous list and the new "list", updates its list and notifies interested listeners of any changes. None of `Category::addItem()`, `Category::changeItem()`, `Category::removeItem()` or `Category::replaceItemList()` should be called prior to the call to `Category::startMonitor()`.

A subclass typically makes zero or more `Category::addItem()` and `Category::setAttr()` calls,

followed by a `Category::endExists()` call followed by zero or more `Category::addItem()`, `Category::changeItem()`, `Category::removeItem()` and `Category::setAttr()` calls.

A Category subclass can also inform interested listeners of application-specific error notifications using `Category::notifyError()`. Error notifications are passed to **CategoryErrorListener** instances that are registered via `Category.addErrorListener()`.

The anticipated use of information in Category is by the client side code. Thus, the Java implementation of the CategoryListener interface is covered in detail in the Obtaining information about Category(s) and Item(s) on the client-side section. The information in that section can be applied to server-side components requiring information from a Category via the C++ CategoryListener API.

Plug-in a Category into the Rhino infrastructure

CategoryFactory is the factory class for Category objects. CategoryFactory methods are mostly used by the **Category Service**, described in `sysadmd(1M)`, to fulfill requests from remote clients. They can also be used by any server-side components that require information from a Category. Category subclasses use the macros defined by CategoryFactory.h.

The steps required to make information about a Category of selector *catName* available to the rest of the system are detailed below. The Category instance will hold information about Item(s) of type *catName*

1. Implement a subclass of Category called *catName*.
 1. The subclass must have a void constructor. Typically, this calls the Category base class constructor with the argument *catName*.
 2. Use the convenience macro `SaCATEGORY_REF_DECL`, provided by CategoryFactory.h, in the header file to provide declaration for the routines used by the Category Service for obtaining Category instances.
 3. In most cases, only one instance of a particular subclass of Category should exist in an address space. To enforce this, subclasses should protect their constructors and use the convenience macro `SaCATEGORY_FRIEND_DEF`, provided by CategoryFactory.h, in the class declaration in the header file. This allows CategoryFactory access to the protected constructors.
 4. Use the convenience macro `SaCATEGORY_REF_DEF`, provided by CategoryFactory.h, in the c++ file to provide the definition for the routines used by the Category Service for obtaining Category instances. This in turn will use the void constructor.
2. Create a library called *catName.so*.
3. Install it in `/usr/sysadm/category/`

The above steps will allow clients to obtain the Category instance for *catName*. To avoid Category name clashes, applications should attach a product specific prefix to their categories. For example, FailSafe Manager and Miser Manager can use a category by name *ResourceCategory*, to refer to different entities. To avoid name clashes, the two categories could be named *fsmgrResourceCategory* and *msmgrResourceCategory*.

Consider, for example, the *UserAccountCategory* Category. In order to plug-in this Category into the Rhino infrastructure, create a library with the naming convention *UserAccountCategory.so*, with the entry points described above and install it in `/usr/sysadm/category`. The Category Service responds to client requests for a Category with selector *UserAccountCategory* by interfacing with *UserAccountCategory.so* to obtain information about the user account Item(s) and passes this

information to the clients.

Code Snippet

Consider a Category named *rhexampRhinoExampleCategory* which is a collection of Item(s), one Item for each file in a particular directory. The Attributes of each Item correspond to the file name, file permissions and contents of the file. The implementation of this Category uses the **fam**(1M) API for obtaining the existing Item(s) in the system and monitoring of future changes is given below.

Header File: *rhexampRhinoExampleCategory.h*

```
#pragma once

#include <sys/types.h>

#include <sysadm/fam.h>

#include <sysadm/Category.h>
#include <sysadm/CategoryFactory.h>

namespace rhexamp {

using namespace sysadm;

SaCATEGORY_REF_DECL(rhexampRhinoExampleCategory);

//
// rhexampRhinoExampleCategory maintains an Item for each known
// RhinoExample.
//
class rhexampRhinoExampleCategory : public Category {

protected:

    rhexampRhinoExampleCategory();
    virtual ~rhexampRhinoExampleCategory();

    // Start monitoring the system.
    virtual void startMonitor();

    // Allow CategoryFactory to create us.
    SaCATEGORY_FRIEND_DEF(rhexampRhinoExampleCategory);

private:

    // Intentionally undefined.
    rhexampRhinoExampleCategory(const rhexampRhinoExampleCategory&);
    rhexampRhinoExampleCategory& operator=(const rhexampRhinoExampleCategory&);

    Item createItem(const char* exampleName);
    void processFamEvent(FAMEvent& event);

    static void famInput(void* clientData, int id, int fd);

    int _inputId;
    FAMConnection _famConn;
    FAMRequest _configDir;
```

```

    bool _famStarted;
};

} // namespace rhexamp

```

C++ File: rhexampRhinoExampleCategory.cpp

```

#include <sys/stat.h>
#include <assert.h>
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>

#include <sysadm/format.h>
#include <sysadm/AppContext.h>
#include <sysadm/Log.h>

#include <rhexamp/RhinoExample.h>
#include "rhexampRhinoExampleCategory.h"

namespace rhexamp {

SaCATEGORY_REF_DEF(rhexampRhinoExampleCategory);

//
// Constructor.
//
rhexampRhinoExampleCategory::rhexampRhinoExampleCategory()
: Category("rhexampRhinoExampleCategory")
{
}

//
// Destructor.
//
rhexampRhinoExampleCategory::~~rhexampRhinoExampleCategory()
{
    if (_famStarted) {
        FAMClose(&_famConn);
        AppContext::getAppContext().unregisterMonitor(_inputId);
    }
}

//
// Item rhexampRhinoExampleCategory::createItem(const char *exampleName)
//
// Description:
//     Respond to a FAM event that indicated that a new example has
//     been created. This also gets called at startup for each
//     example that already existed.
//
//     Make sure example is valid, and if so put together Example
//     attributes.
//
// Parameters:
//     exampleName Name of the example that showed up.
//

```

```

// Returns:
//     Newly created item corresponding to "exampleName".
//
Item rhexampRhinoExampleCategory::createItem(const char *exampleName)
{
    char exampleFile[PATH_MAX];
    (void)SaStringFormat(exampleFile, sizeof exampleFile,
        "%s/%s", RHINO_EXAMPLE_DIR, exampleName);

    struct stat f;
    if (stat(exampleFile, &f) == -1) {
        return Item("", "");
    }

    FILE *fp = fopen(exampleFile, "r");
    if (fp == NULL) {
        return Item("", "");
    }

    char buf[100];
    char *type = fgets(buf, sizeof buf, fp);
    (void)fclose(fp);

    if (type == NULL) {
        return Item("", "");
    }

    char *pc = strchr(buf, '\n');
    if (pc) {
        *pc = '\0';
    }

    Item item(getSelector(), exampleName);
    item.setAttr(Attribute(RHINO_EXAMPLE_NAME, exampleName));
    item.setAttr(Attribute(RHINO_EXAMPLE_TYPE, type));
    item.setAttr(Attribute(RHINO_EXAMPLE_MODE, (long long)f.st_mode));

    return item;
}

//
// void rhexampRhinoExampleCategory::processFamEvent(FAMEvent &event)
//
// Description:
//     Process a single FAM event.
//
// Parameters:
//     event The event to process.
//
void rhexampRhinoExampleCategory::processFamEvent(FAMEvent &event)
{
    Log::trace(getSelector(), "Got a fam event");
    switch (event.code) {
    case FAMExists:
    case FAMCreated:
        {
            Item item(createItem(event.filename));
            if (item.getSelector() != "") {
                addItem(item);
            }
        }
    }
}

```

```

        }
        break;
    case FAMChanged:
    {
        Item item(createItem(event.filename));
        if (item.getSelector() != "") {
            changeItem(item);
        }
    }
    break;
    case FAMDeleted:
        removeItem(event.filename);
        break;
    case FAMEndExist:
        endExists();
        break;
    }
}

//
// void rhexampRhinoExampleCategory::famInput(void* clientData, int, int)
//
// Description:
//     Input callback that gets called when we get a FAM event.
//
// Parameters:
//     clientData  ClusterCategory* (this is a static method).
//
void rhexampRhinoExampleCategory::famInput(void* clientData, int, int)
{
    rhexampRhinoExampleCategory* self = (rhexampRhinoExampleCategory*)clientData;

    FAMEvent event;
    while (FAMPending(&self->_famConn) == 1) {
        if (FAMNextEvent(&self->_famConn, &event) != -1) {
            self->processFamEvent(event);
        }
    }
}

//
// void rhexampRhinoExampleCategory::startMonitor()
//
// Description:
//     Set up our FAM connection.
//
// Returns:
//     0 if successful, -1 if error.
//
void rhexampRhinoExampleCategory::startMonitor()
{
    if (FAMOpen(&_famConn) == 0) {
        _famStarted = true;
        FAMMonitorDirectory(&_famConn, RHINO_EXAMPLE_DIR,
                           &_configDir, NULL);
        _inputId = AppContext::getAppContext().registerMonitor(
            FAMCONNECTION_GETFD(&_famConn), famInput, this);
    } else {
        endExists();
    }
}

```

```
}  
  
} // namespace rhexamp
```

rhexampRhinoExampleCategory.c++ is compiled into a library called rhexampRhinoExampleCategory.so and installed in /usr/sysadm/category. This makes information about rhexampRhinoExampleCategory and its Item(s) available to the rest of the Rhino infrastructure.

Obtaining information about Category(s) and Item(s) on the client-side: Java

All application-specific entities are instances of Item. Further, all application-specific categories are instances of Category. No subclassing is required by the developer of specific application. The steps in obtaining information about Category(s) and Item(s) are:

1. Obtaining a handle to a Category instance.
2. Obtaining information about Item instances.

Obtaining a handle to a Category instance

Category instances are obtained via a **HostContext** object. When writing Task UI interface, a HostContext object will be available for you from the Task infrastructure. The same applies to writing an ItemView, etc.

Internally, the HostContext object is obtained when a user successfully logs in to a server machine.

If the HostContext object is *hostContext*, then a client can obtain a handle to "UserAccountCategory" by using the following code:

```
Category cat = hostContext.getCategory("UserAccountCategory");
```

This is an asynchronous call that returns an handle to the Category before it receives a response from the server. The client can use this handle to add CategoryListener instances for obtaining information. If an error is encountered in communication with the server or loading the specific Category instance requested, this is handled as a fatal connection error by the infrastructure and the client will exit after the error message is acknowledged by the user.

Obtaining information about Items

Clients interested in information about a Category instance can create a subclass of CategoryListener and register for notifications by passing a CategoryListener instance to Category.addCategoryListener(). The specific Items of interest are indicated by the NotificationFilter parameter. The NotificationFilter also specifies whether the CategoryListener instance is interested in notifications about the Category attributes. Call Category.removeCategoryListener() to unregister interest in notification.

Category base class notifies registered CategoryListener instances about Item(s) discovered (by subclasses) in the system or Item(s) that are later added via CategoryListener.itemAdded() calls,

Item changes via `CategoryListener.itemChanged()` calls, and Item removal via `CategoryListener.itemRemoved()`. Notifications about Category Attribute(s) discovered (by subclasses) in the system or Attribute(s) that are later added are via `AttrListener.attrAdded()` calls, Attribute changes via `AttrListener.attrChanged()` calls, and Attribute removal via `AttrListener.attrRemoved()`. When `Category.addCategoryListener()` is called, Category sends the listener its current list of Item(s) and Attribute(s) via `CategoryListener.itemAdded()` and `AttrListener.attrAdded()` calls. End of notification of the current state is signaled by a `CategoryListener.endExists()` call, if the Category itself has received this notification from its subclasses. Else, `CategoryListener.endExists()` will be called when Category receives this notification from its subclasses.

A `CategoryListener` can expect to receive zero or more `itemAdded()` and `attrAdded()` calls, followed by an `endExists()` call followed by zero or more `addItem()`, `changeItem()`, `removeItem()`, `attrAdded()`, `attrChanged()` and `attrRemoved()` calls. The `endExists()` call signals that the Category has communicated the entire set of Item(s) discovered in the system to the `CategoryListener`.

Category base class passes `Category.beginBlockChanges()` and `Category.endBlockChanges()` notifications to identically named methods on registered `CategoryListener` instances.

The following code illustrates how information can be obtained about Category(s) and Item(s) on the client-side. A **CategoryAdapter** is a default implementation of `CategoryListener`. The code below can be used in the Create A User Account Task to verify that the user account name does not already exist on the server.

```
Category cat = hostContext.getCategory("UserAccountCategory");
cat.addCategoryListener(
    new CategoryAdapter() {
        public void itemAdded(Item item) {
            if (item.getString(NAME).equals(userInputName)) {
                // UserAccount with name userInputName already exists
                // Steps to signal error ...
            }
        }
        public void endExists() {
            // UserAccount with name userInputName does not exist
            // Steps to signal successful verification ...
        }
    }, NotificationFilter.ALL_ITEMS);
```

The above method can be used when the user account name is not the same as the selector of the Item. If the client has the selector (unique name) of the Item, then there are two other ways of determining if the user account that was specified already exists. The following code uses a `CategoryListener` with a `NotificationFilter` that expresses interest in only one Item with a specified selector. If an Item with that selector exists, Category will pass the Item state to the `CategoryListener` via `CategoryListener.itemAdded()` followed by an `CategoryListener.endExists()`. If an Item with that selector does not exist it will send an `CategoryListener.endExists()` notification.

```
NotificationFilter filter = new NotificationFilter();
filter.monitorItem(selector);

cat.addCategoryListener(
    new CategoryAdapter() {
        public void itemAdded(Item item) {
```

```

        // UserAccount with unique name "selector" already exists
        // Steps to signal error ...

    }
    public void endExists() {
        // UserAccount with unique name "selector" does not exist
        // Steps to signal successful verification ...
    }
}, filter);

```

Another way of obtaining an Item if the client has the selector is to use the Category.getItem() API, passing a **ResultListener** to Category.getItem(). Category.getItem() calls the succeeded method of the ResultListener if an Item with the specified selector exists in the system. Use the getResult() method of **ResultEvent** to get the Item. The Object returned by getResult() should be cast to an Item. getItem() calls the failed() method of the ResultListener if an Item with the specified selector does not exist in the system.

```

cat.getItem(selector, new ResultListener() {
    public void succeeded(ResultEvent event) {
        // UserAccount with unique name "selector" already exists
        // Steps to signal error ...
    }

    public void failed(ResultEvent event) {
        // UserAccount with unique name "selector" does not exist
        // Steps to signal successful verification ...
    }
});

```

Similar to Category.getItem(), Category.getItemCount() can be used to get the number of Items in a Category and Category.getItemList() can be used to get the list of Items in a Category.

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

Related: [Basic Concepts](#) | [Architecture](#) | [Item & Category](#)

Association in Rhino

An **Association** represents a relationship between an Item and other Item(s) in the same or different Category. For example, the user account Item(s) that are related to a group account Item with selector "bar" could be represented by an Association. The identifying parameters for each Association are a parent Category, selector of an Item in that Category (parent Item) and a child Category. An Association represents an 1-to-n relationship. A 1-to-1 relationship is handled as a special case of 1-to-n relationship.

Association derives from Category and supports the same client mechanisms, using CategoryListener interface, for obtaining dynamic information about the Item(s) in an Association instance. The list of Item(s) in an Association is of the same type as indicated by the (selector of) child Category. An Association can have a set of Attributes that apply to all Item(s) in that relationship. The Association class provides API to support monitoring of Item(s) and Attribute(s) in the specific relationship and notification of current state and state changes to interested clients.

If an Item with the specified parent selector does not exist, the Association will monitor the parent Category for the addition of an Item with that selector. When such an Item is added, the subclasses are notified and subclasses can do whatever is necessary to determine the child Item(s) and Association Attribute(s) and monitor the system for future changes.

If an Item with the specified parent selector existed, but is deleted while a client is monitoring the Association, all Items are removed from the Association and the Association continues monitoring of the parent Category for the addition of an Item with that parent selector.

The concept of an Association is implemented in Java at the client-side and in C++ at the server-side. An application developer writes the logic to determine the set of Item(s) related to a parent Item and Association Attribute(s) and to monitor the system for any changes to the Item(s) and Association Attribute(s) in C++, using the server-side API. The application-specific clients, such as specific Tasks, ItemViews and TreeView are written in Java and use the client-side API to obtain information about Association(s) and Item(s).

Implementing Association(s) on the server-side: C++

The Association class and three subclasses **ComputedAssoc**, **ChildAttrAssoc** and **ParentAttrAssoc** provide support for implementing Association(s) on the server-side.

Association

The most basic support for specifying association is provided by the Association class. This class monitors the parent Item and subclasses are responsible for determining the child Item(s) that added, deleted and changed. The Association class (versus derived classes) is used only when sub-classes require logic very specific to the application to determine when items are added, changed and removed from the association. For example, the relationship may be calculated by queries on a database based on the parent Selector. The derived classes **ComputedAssoc**, **ChildAttrAssoc** and **ParentAttrAssoc** is used

when the data requiring for computing the relationship is contained within the Item(s) in the Category (s).

Infrastructure Support for Association

```
// Provided by infrastructure
class Category : public AttrBundle {
    ...
    virtual void addItem(const& Item item);
    virtual void removeItem(const& Item item);
    virtual void changeItem(const& Item oldItem, const& Item newItem);
    ...
};

// Provided by infrastructure
class Association : public Category {
    ...
    Association(Category& parentCategory, const String& parentSelector,
                Category& childCategory);

    // Association interacts with Category with selector parentCategory
    // and calls the following methods when the parent Item with
    // selector parentSelector is added, changed or removed.
    virtual void parentAdded(const Item& item);
    virtual void parentChanged(const Item& oldItem, const Item& newItem);
    virtual void parentRemoved();
    ...
};
```

Using Association

```
// Written by application developer
class MachinesAssocWithCluster : public Association {
    MachinesAssocWithCluster(Category& parentCategory,
                              const String& parentSelector,
                              Category& childCategory) :
        Association(parentCategory, parentSelector, childCategory);
    virtual void parentAdded(const Item& item);
    virtual void parentChanged(const Item& oldItem,
                               const Item& newItem);
};
```

The Association class provides trivial implementations of `Association::parentAdded()` and `Association::parentChanged()`. Subclasses can override `Association::parentAdded()` to, for example, register for notifications about Items in the child Category. Subclasses can override `Association::parentChanged()` to do whatever is necessary to keep the list of child Item(s) and Association Attribute(s) up-to-date. The Association class implements `Association::parentRemoved()` to remove all Item(s) from its list and notify removal of Item(s) to registered listeners. Most of the rules of subclass interaction with the Category base class apply. The differences are:

- `Category::startMonitor()` is overridden by the Association base class to start monitoring of the parent Item. The Association class will call `Association::parentAdded()` if the parent Item is determined to exist upon start of monitoring or added later. Subclasses of Association should not call `Category::addItem()`, `Category::changeItem()`, `Category::removeItem()`, `Category::replaceItemList()`, `AttrBundle::setAttr()` prior to the call to

```
Association::parentAdded().
```

- The selector of the Association instance is not the same as the type Item(s) in its list. Item(s) in the list are of type corresponding to the selector of the child Category.

Typically a subclass makes zero or more `Category::addItem()` and `AttrBundle::setAttr()` calls, followed by a `Category::endExists()` call followed by zero or more `Category::addItem()`, `Category::changeItem()`, `Category::removeItem()` and `Category::setAttr()` calls. The `Category::endExists()` call signals that the subclass has communicated the entire set of Item(s) and Association attributes discovered in the system to the Category base class.

The anticipated use of information in Association is by the client side code. Clients can access Association information using the CategoryListener interface in the same manner as they would obtain information from a Category. The only difference is the call to obtain the handle to an Association. This is covered in detail in the [Obtaining information about Association\(s\) on the client-side](#) section. The information in that section can be applied to server-side components requiring information from an Association via the C++ CategoryListener API.

ComputedAssoc

This is the base class for deriving classes to represent relationships that can be computed from values of Attribute(s) of the monitored parent and child Item(s). This class monitors the parent Item from the parent Category and Item(s) in the child Category that are potential children. The Item(s) that are potential children are indicated by a NotificationFilter that comes in effect when the parent Item is detected by Association. The NotificationFilter can be changed when the parent Item changes or at any arbitrary time.

Infrastructure Support for ComputedAssoc

```
// Provided by infrastructure
class ComputedAssoc : public Association {
...
// Monitoring of potential child Item(s)
virtual void childCategoryItemAdded(const Item& item);
virtual void childCategoryItemChanged(const Item& oldItem,
                                     const Item& newItem);
virtual void childCategoryItemRemoved(const String& selector);
...
// Set NotificationFilter to indicate potential child Item(s)
virtual NotificationFilter*
    createAddedChildNotificationFilter(const Item& parentItem);
virtual NotificationFilter*
    createChangedChildNotificationFilter(const Item& oldItem,
                                       const Item& newItem);
virtual void adoptAndReplaceChildNotificationFilter(NotificationFilter*
                                                    filter);
...
//
...
virtual bool isChild(const Item& potentialChildItem) = 0;
};
```

Subclasses must override the `ComputedAssoc::isChild()` method to provide logic that determines if the Item of the child Category passed to `ComputedAssoc::isChild()` is a child of the parent Item.

This class marks all Item(s) of the child Category as potential child Item(s). This class calls `ComputedAssoc::isChild()` anytime there is a change to the parent or the monitored child Item(s) and forwards notifications about changes in its list of child Item(s) to the listeners on this Association. When the parent Item changes, it gets the current list of child Item(s) and checks if any child Item(s) need to be added/removed from its list based on the computation performed by `ComputedAssoc::isChild()`. When an Item of the child Category is added, removed or changed, it checks if the Item should be added/removed/updated in its list of child Item(s) based on `isChild()`.

For example, the user account Item(s) belonging to a particular group account Item could be stated as the user account Item(s) that have the same value of Attribute with key *uid* as the corresponding Attribute in the group account Item. The following code can be used to model this relationship.

Using ComputedAssoc

```
// Provided by application developer
class UsersAssocWithGroup : public ComputedAssoc {
...
    bool isChild(const Item& parentItem, const Item& childItem) {
        if (childItem.getAttr("uid").stringValue() ==
            parentItem.getAttr("uid").stringValue()) {
            return true;
        } else {
            return false;
        }
    }
...
}
```

Sub-classes can further fine-tune behaviour by overriding the methods that are called upon notifications related to the parent and child Item(s). For example, subclass `ChildAttrAssoc` overrides `parentChanged()` to turn off computation of the list of child Item(s) when the parent Item changes and only does so on changes in the Item(s) of the child Category.

ChildAttrAssoc

This is the base class for deriving classes to represent relationships in which the child Item(s) store the selectors of one or more parent Item(s) as part of its Attributes.

Subclasses provide the Attribute's key in the constructors. The `ParentAttrAssoc` class monitors the parent and child Item(s), keeps the list of child Item(s) current and notifies listeners of changes to the list of child Item(s).

`ParentAttrAssoc` can determine the Item(s) of child category that belong to this Association in two ways depending on the constructor that is used. One constructor is used when there is a 1-to-1 relationship from a child Item to Item(s) of the parent Category. Subclasses specify the Attribute key of the child Item that holds the parent Item selector. Another constructor is used when there is a 1-to-n relationship from a child Item to Item(s) of the parent Category. This is based on the recommended format for representing arrays of values in an Item. Subclasses specify the value of the key of the child Item that holds the number of parent Item selectors and the base name of the Attribute keys which hold the selectors themselves.

For example, if `ClustersAssocWithMachine` is a relationship where an Item of type `ClusterCategory` has

an array of selectors of Item(s) of MachineCategory, NUM_MACHINES could be the key of ClusterCategory Item(s) that holds number of values in the array. The base name of the attribute keys could be MACHINE. ParentAttrAssoc will monitor Item(s) corresponding to the list of selectors in Attributes of the child ClusterCategory Item(s) with keys CLUSTER0, CLUSTER1, ... CLUSTER<NUM_CLUSTERS - 1>.

ChildAttrAssoc keeps the list of child Item(s) up-to-date based on parent/child changes.

Infrastructure Support for ChildAttrAssoc

```
// Provided by infrastructure
class ChildAttrAssoc : public ComputedAssoc {
...
// Constructor. This version is used for 1-to-n relationships and
// takes two attribute names.
// "childAttrNumKeys" is the attribute key which holds the number of
// parent Item selectors, and "childAttrKey" is the base name of
// the attribute keys which hold the selectors themselves.
ChildAttrAssoc(Category& parentCategory,
               const String& parentSelector,
               Category& childCategory,
               const String& childAttrNumKeys,
               const String& childAttrKey);

// Constructor. This version is used for 1-to-1 relationships and
// takes one attribute name which is the attribute key holding the
// parent Item selector.
ChildAttrAssoc(Category& parentCategory,
               const String& parentSelector,
               Category& childCategory,
               const String& childAttrKey);
...
}
```

Typically, subclasses only need to call the ChildAttrAssoc constructor with the application specific values of the Attribute keys and do not need to override any methods. For example, if a user account Item has an attribute with key "groupSelector" that refers to the parent group account Item, the following code can be used to model this relationship.

Using ChildAttrAssoc

```
// Provided by application developer
class UsersAssocWithGroup : public ChildAttrAssoc {
    UsersAssocWithGroup(Category& parentCategory,
                       const String& parentSelector,
                       Category& childCategory) :
        ChildAttrAssoc(parentCategory, parentSelector, childCategory,
                       "groupSelector") {
    }
};
```

ParentAttrAssoc

This is the base class for deriving classes to represent relationships in which the parent Item stores the

selectors of one or more child Item(s) as part of its Attributes.

Subclasses provide the Attribute's key in the constructors. The ParentAttrAssoc class monitors the parent and child Item(s), keeps the list of child Item(s) current and notifies listeners of changes to the list of child Item(s).

ParentAttrAssoc can determine the Item(s) of child category that belong to this Association in two ways depending on the constructor that is used. One constructor is used when there is a 1-to-1 relationship from a parent Item to Item(s) of the child Category. Subclasses specify the Attribute key of the parent Item that holds the child Item selector. Another constructor is used when there is a 1-to-n relationship from a parent Item to Item(s) of the child Category. This is based on the recommended format for representing arrays of values in an Item. Subclasses specify the value of the key of the parent Item that holds the number of child Item selectors and the base name of the Attribute keys which hold the selectors themselves.

For example, if ClustersAssocWithMachine is a relationship where an Item of type MachineCategory has an array of selectors of Item(s) of ClusterCategory, NUM_CLUSTERS could be the key of MachineCategory Item(s) that holds number of values in the array. The base name of the attribute keys could be CLUSTER. ParentAttrAssoc will monitor Item(s) corresponding to the list of selectors in Attributes of the parent MachineCategory Item with keys CLUSTER0, CLUSTER1, ... CLUSTER<NUM_CLUSTERS - 1>.

ParentAttrAssoc keeps the list of child Item(s) up-to-date based on parent/child changes.

Infrastructure Support for ParentAttrAssoc

```
// Provided by infrastructure
class ParentAttrAssoc : public ComputedAssoc {
...
// Constructor. This version is used for 1-to-n relationships and
// takes two attribute names.
// "parentAttrNumKeys" is the attribute key which holds the number of
// child Item selectors, and "parentAttrKey" is the base name of
// the attribute keys which hold the selectors themselves.
ParentAttrAssoc(Category& parentCategory,
                const String& parentSelector,
                Category& childCategory,
                const String& parentAttrNumKeys,
                const String& parentAttrKey);

// Constructor. This version is used for 1-to-1 relationships and
// takes one attribute name which is the attribute key holding the
// child Item selector.
ParentAttrAssoc(Category& parentCategory,
                const String& parentSelector,
                Category& childCategory,
                const String& parentAttrKey);
...
};
```

Typically, subclasses only need to call the ParentAttrAssoc constructor with the application specific values of the Attribute keys and do not need to override any methods. For example, if a group account Item has an attribute with key "NUM_USERS" that specifies the number of user account Item(s) belonging to it, USER0, USER1 ... USER<NUM_USERS - 1> would be the keys of the actual selector

values, the following code can be used to model this relationship.

```
// Provided by application developer
class UsersAssocWithGroup : public ParentAttrAssoc {
    UsersAssocWithGroup(Category& parentCategory,
                        const String& parentSelector,
                        Category& childCategory) :
        ParentAttrAssoc(parentCategory, parentSelector, childCategory,
                        "NUM_USERS", "USER") {
    }
};
```

Plug-in an Association into the Rhino infrastructure

AssocFactory is the factory class for Association objects. AssocFactory methods are used by the **Association Service** (similar to the Category Service described in sysadmd(1M)) to fulfill requests from remote clients. They can also be used by any server-side components that require information from an Association. Association subclasses use the macros defined by AssocFactory.h.

To make information about an Association between any Item of type *parentCategorySelector* and Item(s) of type *childCategorySelector* available to the rest of the system the following steps are required:

1. Information about the Category(s) *parentCategorySelector* and *childCategorySelector* should be available as detailed in the document Item and Category in Rhino.
2. Implement a subclass of Association called `<parentCategorySelector>AssocWith<childCategorySelector>`.
 1. The subclass must have a constructor with the signature (Category& parentCategory, const String& parentSelector, Category&). Typically, this calls the corresponding Association constructor or any Association subclass constructor. *parentSelector* is the name of the parent Item in a Category instance *parentCategory* corresponding to *parentCategorySelector* for which related Item(s) from Category instance *childCategory* corresponding to *childCategorySelector* are to be determined.
 2. Use the convenience macro `SaASSOC_REF_DECL`, provided by AssocFactory.h, in the header file to provide declaration for the routines used by the Association Service for obtaining Association instances.
 3. In most cases, only one instance of a particular subclass of Association should exist for a given "parentCategory", "parentSelector", "childCategory" combination in an address space. To enforce this, subclasses should protect their constructors and use the convenience macro `SaASSOC_FRIEND_DEF`, provided by AssocFactory.h, in the class declaration in the header file. This allows AssocFactory access to the protected constructors.
 4. Use the convenience macro `SaASSOC_REF_DEF`, provided by AssocFactory.h, in the c++ file to provide the definition for the routines used by the Association Service for obtaining Association instances. This in turn will use the mandatory constructor described.
3. Create a library called `<parentCategorySelector>AssocWith<childCategorySelector>.so`.
4. Install it in `/usr/sysadm/association/`

The above steps will allow clients to obtain the Association instance to determine the relationship between any Item of type *parentCategorySelector* and Item(s) of type *childCategorySelector*.

Obtaining information about Association(s) on the client-side: Java

All application-specific entities are instances of `Item`. Further, all application-specific associations are instances of `Association`. No subclassing is required by the developer of specific application. The steps in obtaining information about `Association(s)` and `Item(s)` are:

- Obtaining a handle to an `Association` instance.
- Obtaining information about `Item` instances.

Obtaining a handle to an `Association` instance

`Association` instances are obtained via a **`HostContext`** object. When writing Task UI interface, a `HostContext` object will be available for you from the Task infrastructure. The same applies to writing an `ItemView` etc.,.

Internally, the `HostContext` object is obtained when a user successfully logs in to a server machine.

If the `HostContext` object is *hostContext*, then a client can obtain a handle to an `Association` representing the user account `Item(s)` belonging to a parent group account `Item` called *foo* by using the following code:

```
Association assoc =  
    hostContext.getCategory("GroupAccountCategory", "foo",  
                           "UserAccountCategory");
```

This is an asynchronous call that returns an handle to the `Association` before it receives a response from the server. The client can use this handle to add `CategoryListener` instances for obtaining information. If an error is encountered in communication with the server or loading the specific `Association` instance requested, this is handled as a fatal connection error by the infrastructure and the client will exit after the error message is acknowledged by a user.

Obtaining information about `Items`

All details of Obtaining information about `Items` from a category apply to obtaining information from an `Association`.

Copyright (c) 1999, 2000 Silicon Graphics, Inc. All Rights Reserved.

[Overview](#) | [Basic Concepts](#) | [GUI Components](#) | [Architecture](#) | [How To Write An App](#)

How to Write a Task

Outlined below are the basic steps involved in writing a Task for Rhino. Unless otherwise noted, the code examples below are for a Define User Account Task.

Before proceeding, you should familiarize yourself with [Basic Concepts](#) and at least look over the [GUI Components](#), [Architecture](#), and [Task Internals](#) documents.

1. Create the Task properties file

The Task properties file is required to exist in the same directory as the Task class. The name of the properties file must be the Task class name followed by "P.properties". For example, a Task subclass named "DefineUserAccountTask" must have a properties file named "DefineUserAccountTaskP.properties".

The Task properties file contains static information about the Task, including the Task title, privilege list, and whether or not the Task accepts operands. The Task properties file also contains User-visible labels and messages, and interface characteristics such as fonts, colors, and sizes.

Below is a sample of what the DefineUserAccountTaskP.properties file might contain. Letters have been used to identify the lines to distinguish them from the Java code examples that follow.

```
A: #
B: # Properties file for the Define User Account Task
C: #
D: Task.shortName = Define User Account
E: Task.longName = Define a new User Account
F: Task.keywords = define new add user account login home directory shell
G:
H: [...]
I:
J: #
K: # DO NOT LOCALIZE BELOW THIS LINE
L: #
M: Task.privList0 = addUser
N: Task.privList1 = listUsers
O: Task.publicData0 = userName
P: Task.ProductAttributes0 = com.sgi.psa
```

2. Implement the Task subclass

The Task subclass is the main entry point to the Task. Its functions are to verify prerequisites, initialize TaskData, coordinate the Task interface(s), and perform the Task operation when the User presses the OK button.

- **Constructor**

The Task subclass constructor is responsible for initializing all public and private TaskData and

adding TaskDataVerifiers for each TaskData attribute that needs to be verified when the User presses OK.

```
1: private TaskContext _taskContext;
2: private TaskData _taskData;
3: private ResourceStack _rs;
4: private HostContext _hc;
5: private Category _userCategory;
6:
7: private static final String USER_NAME = "userName";
8:
9: public DefineUserAccountTask(TaskContext taskContext) {
10:     super(taskContext);
11:
12:     _taskContext = taskContext;
13:     _taskData = _taskContext.getTaskData();
14:     _rs = _taskContext.getResourceStack();
15:     _hc = _taskContext.getHostContext();
16:
17:     _taskData.setString(USER_NAME, "");
18:     _taskContext.appendTaskDataVerifier(USER_NAME, new
19:                                     TaskDataVerifier() {
20:         public void dataOK(int browseFlag, Object context,
21:                             ResultListener listener) {
22:             verifyUserName(browseFlag, context, listener);
23:         }
24:     });
25:
26:     [...]
27: }
28:
29: public void verifyUserName(int browseFlag, Object context,
30:                             ResultListener listener) {
31:     String userName = _taskData.getString(USER_NAME);
32:     ResultEvent result = new ResultEvent(this);
33:
34:     if (userName.length() == 0) {
35:         if (browseFlag) {
36:             listener.succeeded(result);
37:         } else {
38:             result.setReason(
39:                 _rs.getString("Error.missingUserName"));
40:             listener.failed(result);
41:         }
42:     }
43:
44:     // Check user name for syntactic problems (length,
45:     // unprintable characters, etc.).
46:     [...]
47:
48:     // Check for existing user name
49:     verifyUniqueName(userName, listener);
50: }
51:
52: public void verifyUniqueName(final String userName,
53:                             final ResultListener listener) {
54:     _userCategory = _hc.getCategory("UserAccountCategory");
55:     _userCategory.getItem(userName, new ResultListener() {
56:         public void succeeded(ResultEvent event) {
```

```

57:         event.setReason(MessageFormat.format(
58:             _rs.getString("Error.userExists"),
59:             new Object[] { userName } ));
60:         listener.failed(event);
61:     }
62:
63:     public void failed(ResultEvent event) {
64:         listener.succeeded(event);
65:     }
66: });
67: }

```

- **Task.registerInterfaces()**

Task.registerInterfaces() is abstract, therefore it must be implemented by the subclass. Its responsibilities are to:

1. Create the Task interface classes (Form and/or Guide) and register those classes with the Task base class using Task.setForm() and Task.setGuide() respectively.
2. Override the default title string if it is to include something other than *Task.shortName* and the server name (not shown in the example below).

```

68:     public void registerInterfaces() {
69:         setForm(new DefineUserAccountForm(_taskContext));
70:         setGuide(new DefineUserAccountGuide(_taskContext));
71:     }

```

- **Task.setOperands()**

Task.setOperands() is only required if the property Task.operandTypeAccepted is set. Note that setOperands() is synchronous, so it should not do any operand verification that involves a server request. Server verification should be deferred until verifyPrereqsBeforeCheckPrivs() or verifyPrereqsAfterCheckPrivs().

Although the DefineUserAccountTask does not take operands, below is some sample code for the ModifyUserAccountTask, which accepts a single User Account as an operand.

```

72:     public void setOperands(vector Operands)
73:         throws TaskInitFailedException {
74:         if (operands == null || operands.size() == 0) {
75:             // The operands are optional
76:             return;
77:         }
78:
79:         if (operands.size() > 1) {
80:             throw new TaskInitFailedException(
81:                 _rs.getString("Error.tooManyOperands"),
82:                 TaskInitFailedException.INVALID_OPERANDS);
83:         }
84:
85:         // In the future, operands could be Items that are
86:         // dropped via the drag and drop interface. In
87:         // Rhino 1.0, however, we can only pass String operands
88:         // at this time.
89:         if (!(operands.elementAt(0) instanceof String)) {
90:             throw new TaskInitFailedException(
91:                 _rs.getString("Error.invalidOperandType"),
92:                 TaskInitFailedException.INVALID_OPERANDS);

```

```

93:         }
94:
95:         // Store the operand for later use
96:         _taskData.setString(USER_NAME, (String)operands.elementAt(0));
97:     }

```

- **Task.verifyPrereqsBeforeCheckPrivs()**

Task.verifyPrereqsBeforeCheckPrivs() is provided as a hook for subclasses to do whatever verification is possible before privileges are obtained. The base class provides a trivial implementation of verifyPrereqsBeforeCheckPrivs() that always succeeds, so you do not need to implement this method if you have no prerequisites to verify or if all of your checking requires privileges.

For illustrative purposes, imagine that the Define User Account Task allows the proposed User name be set via setTaskDataAttr() (note that "userName" was declared as a public TaskData attribute in the Task properties file above). A prerequisite could be that the User name may not already exist on the server. Below is the code that would be used to verify the User name as a prerequisite. Note that the "userName" TaskDataVerifier, defined in lines 18-24 of the constructor, is being referenced here by name.

```

98:     public void verifyPrereqsBeforeCheckPrivs(ResultListener listener) {
99:         _taskContext.dataOK(USER_NAME, TaskDataVerifier.MAY_BE_EMPTY,
100:                             null, listener);
101:     }

```

- **Task.verifyPrereqsAfterCheckPrivs()**

Task.verifyPrereqsAfterCheckPrivs() is provided as a hook for Task subclasses that need privileges in order to verify some of their prerequisites. The base class provides a trivial implementation of verifyPrereqsAfterCheckPrivs() that always succeeds, so you do not need to implement this method if you have no privileged verification.

Privileged verification requires a call to the version of Task.runPriv() that takes a ResultListener. A generic example follows.

```

102:     public void verifyPrereqsAfterCheckPrivs(ResultListener listener) {
103:         TaskData CLIArgs = new TaskData();
104:         CLIArgs.setAttr(_taskData.getAttr("prereqData"));
105:         [...]
106:
107:         OutputStream stream = runPriv("listUsers", CLIArgs, listener);
108:
109:         try {
110:             stream.close();
111:         } catch (IOException ex) {
112:             Log.debug("DefineUserAccountTask",
113:                     "unable to close listUsers stream");
114:         }
115:     }

```

- **Task.ok()**

Task.ok() is responsible for sending a request to the server to perform the requested Task. In the most common case, a single privileged command is needed to perform the Task. When the Task is more complex, however, multiple privileged commands may be involved. The sample code below covers the typical case. See the Task API documentation for details on other versions of

`Task.runPriv()` that can be used in more complex Tasks.

```

116:    public void ok() {
117:        TaskData CLIArgs = new TaskData();
118:        CLIArgs.setAttr(_taskData.getAttr(USER_NAME));
119:        [...]
120:
121:        OutputStream stream = runPriv("addUser", CLIArgs);
122:
123:        try {
124:            stream.close();
125:        } catch (IOException ex) {
126:            Log.debug("DefineUserAccountTask",
127:                    "unable to close OutputStream from runPriv");
128:        }
129:    }

```

- **createResultViewPanel()**

Task clients that want to display a `ResultViewPanel` after being notified that the Task has succeeded call `Task.getResultViewPanel()`. That method checks to make sure the Task succeeded and then calls the abstract `Task.createResultViewPanel()`. The `ResultViewPanel` should show information about the object modified or created, if appropriate; text describing the result and consequences of the Task; and a list of Tasks that the User might logically want to invoke next.

```

130:    public ResultViewPanel createResultViewPanel() {
131:        return new ResultViewPanel(_taskContext, _rs,
132:                                   "UserAccountCategory",
133:                                   _rs.getString(
134:                                       "DefineUserAccountTask.succeeded"));
135:    }

```

3. Implement the Form subclass

The Form subclass sets up the visible components for the Form interface of a Task, and then binds the components to the appropriate `TaskData` attributes.

- **Constructor**

The Form subclass constructor has no specific responsibilities. In practice, it is used to create aliases to the `TaskContext` and any other classes that will be shared with the Task subclass.

```

1:public class DefineUserAccountForm extends Form {
2:
3:    private TaskContext _taskContext;
4:    private TaskData _taskData;
5:
6:    public void DefineUserAccountForm(TaskContext taskContext) {
7:        super(taskContext);
8:
9:        _taskContext = taskContext;
10:       _taskData = _taskContext.getTaskData();
11:    }

```

- **Form.createUI()**

`Form.createUI()` is responsible for creating the visible components of the Form interface. It is not called until the Form is actually displayed. `Form.createUI()` should always call `super.createUI()` as

its first act in order to create the Form icon and title at the top of the Form interface. See the API documentation for details.

```

12:     public void createUI() {
13:         super.createUI();
14:
15:         ResourceStack rs = _taskContext.getResourceStack();
16:
17:         FilteredTextField userName =
18:             new FilteredTextField(
19:                 rs.getInt("DefineUserAccountForm.userNameFieldWidth"),
20:                 FilteredTextField.BEEP);
21:
22:         addTaskComponent(userName,
23:             rs.getString(
24:                 "DefineUserAccountForm.userNameLabel"));
25:         StringJTextComponentBinder.bind(_taskData, USER_NAME, userName);
26:
27:         // More visible components would be added here
28:         [...]
29:     }

```

4. Implement the Guide subclass

The Guide subclass is slightly more complicated than the Form interface because the developer breaks the interface into multiple pages, each of which may have its own TaskDataVerifier that gets called when the User presses the Next button to leave that page.

- **Constructor**

The Guide subclass constructor has no specific responsibilities. In practice, it is used to create aliases to the TaskContext and any other classes that will be shared with the Task subclass.

```

1: public class DefineUserAccountGuide extends Guide {
2:
3:     private TaskContext _taskContext;
4:     private TaskData _taskData;
5:     private ResourceStack _rs;
6:
7:     public DefineUserAccountGuide(TaskContext taskContext) {
8:         super(taskContext);
9:
10:         _taskContext = taskContext;
11:         _taskData = _taskContext.getTaskData();
12:         _rs = _taskContext.getResourceStack();
13:     }

```

- **Guide.registerPages()**

Guide.registerPages() is responsible for creating and registering each of the GuidePages that make up the Guide. If your Guide has pages that only appear if the User selects certain options, those pages can be registered later using either Guide.appendPage() or Guide.insertPage(). See the API documentation for more details about creating GuidePages.

The verification for a GuidePage is called when the User presses the Next button. The example below is very simple because there is only one input field on the page. TaskContext has additional versions of the dataOK() method that allow the developer to chain together a set of

TaskDataVerifiers. If a verifier fails, the default action that the Task will take is to post an error dialog containing the reason field of the ResultEvent returned by the verifier. This error dialog will contain two buttons - one to stay on the current page and fix the error, and the other to ignore the error for now and go on to the next page. This second button gives users the ability to see what will appear on subsequent pages without having to fill in the current page. If your task does not support going to the next page until all the verifiers for the current page succeed (for example, if the input on current page determines which page the user sees next), then use GuidePage's `setAllowTurnPageOnError` method to tell the infrastructure to not let the user turn the page until all verifiers on the current page succeed.

```

14:     public void registerPages() {
15:         GuidePage userNamePage =
16:             new GuidePage(_taskContext, "UserNamePage") {
17:                 public void createUI() {
18:                     super.createUI();
19:
20:                     FilteredTextField userName =
21:                         new FilteredTextField(_rs.getInt(
22:                             "DefineUserAccountForm.userNameFieldWidth"),
23:                             FilteredTextField.BEEP);
24:
25:                     addTaskComponent(userName,
26:                         rs.getString("DefineUserAccountForm.userNameLabel")
27:                         StringJTextComponentBinder.bind(_taskData, USER_NAME,
28:                                                         userName));
29:                 }
30:             }
31:
32:         userNamePage.setVerifier(new TaskDataVerifier() {
33:             public void dataOK(final int browseFlag, final Object context,
34:                               final ResultListener listener) {
35:                 _taskContext.dataOK(USER_NAME, browseFlag, context,
36:                                     listener);
37:             }
38:         });
39:
40:         appendPage(userNamePage);
41:
42:         // Additional pages would be added here
43:         [...]
44:     }

```

Running a Task from the command line

```

% setenv CLASSPATH \
/usr/sysadm/java/swingall.jar:/usr/sysadm/java/sysadm.jar:{task workarea}

% java com.sgi.sysadm.manager.RunTask {package}.{taskname} [operands]

```

See the RunTask documentation for a list of available runtime options.

Areas to be covered in a future revision

Integrating Tasks into Other Views

There are two ways that a set of Tasks can be associated with a particular view:

1. The list of Tasks is hardcoded in the view code or properties file.
2. The list of Tasks is retrieved from the TaskRegistry, filtered by ItemTester.

Sharing Code Among Multiple Tasks

1. **Subclassing**
2. **Libraries**

Tips on Asynchronous Programming

- **Why Asynchronous Calls are Needed**
- **Common Problems in Asynchronous Programming**

How to use a Rhino IconRenderer

Table of Contents

- [Introduction](#)
 - [Before you begin](#)
 - [Specifying the IconRenderer to use for a Category](#)
 - [Using the ResourceBasedIconRenderer](#)
 - [Using a subclass of IconRenderer](#)
-

Introduction

This document is a reference for SGI software engineers who will be using IconRenderers for Rhino applications. IconRenderer is a class that can generate an icon that represents a particular Item. Many of the Rhino infrastructure components use an IconRenderer to display the icon associated with an Item, including the ItemView, the ItemTable, the ResultView, and the TreeView. The IconRenderer gives the programmer the ability to define what icon an Item should use, and have that icon used everywhere that an Item's icon is displayed.

There is one IconRenderer for each Category. The IconRenderer is responsible for monitoring the Category and generating icons for any Items that listeners have expressed interest in.

For more information on IconRenderers in general, see the [IconRenderer](#) documentation API documentation.

Before you begin

Before you begin to create an IconRenderer for a particular Category, you need to understand the names and terms that the Rhino infrastructure uses in relation to Categories. See the [The names of Categories on the client and on the server](#) documentation for more information.

Specifying the IconRenderer to use for a Category

The [HostContext](#) keeps track of which IconRenderer to use for each Category. To specify a Category's IconRenderer, place the `ICON_RENDERER` resource in the Category's resource file. For example, to specify that the RhinoExampleCategory should use the "com.sgi.rhexamp.category.IconRenderer" class as its IconRenderer, the following entry would be made

in /com/sgi/rhexamp/category/rhexampRhinoExampleCategoryP.properties:

```
com.sgi.rhexamp.category.rhexampRhinoExampleCategory.iconRenderer = com.sgi.rhexamp.
```

If this resource is not specified, then the HostContext creates a ResourceBasedIconRenderer object for the Category.

Using the ResourceBasedIconRenderer

The ResourceBasedIconRenderer is a subclass of IconRenderer that is used to display icons if there is no `ICON_RENDERER` resource specified (see Specifying the IconRenderer to use for a Category). To use the ResourceBasedIconRenderer, you must place specific resources in the Category's resource file.

Using the same icon for all the Items in a Category

If you want to use the same icon for all the Items in a Category, then only the "icon" resource is needed, defined as `<name>.icon` (see the DEFAULT_ICON documentation for more information). The icon described by the "icon" resource can be either classpath-relative pathnames to .gif or .jpg icons, or they can be package-qualified names of FtrIcon. For example, if all of the Items in the RhinoExample Category should show an icon of a rhino, then the following lines would be placed in the resource file:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B: ${RHINO_EXAMPLE_CATEGORY}.icon = /com/sgi/sysadm/ui/images/sysadm.gif
```

Using different icons for each Item in a Category

To show a different icon for each Item in the Category, add the "iconBasedOn" resource, defined as `<name>.iconBasedOn`, where `<name>` is the package-qualified name of the Category (see the ICON_BASED_ON documentation for more information). This resource specifies which of the Item's Attributes the icon will be based on.

In conjunction with the "iconBasedOn" resource, there should be "icon" resources, defined as `<name>.icon.<Attribute's value>`, where `<Attribute's value>` is one of the values that the Attribute specified by "iconBasedOn" can have. (See the ICON documentation for more information). If no "icon" resource is found that matched the value of the Attribute specified by "iconBasedOn", then the default icon (as described above) will be used.

For example, in the RhinoExample category, the icon is based on the type, so the following entries are made in the resource file:

```
A: RHINO_EXAMPLE_CATEGORY=com.sgi.rhexamp.category.rhexampRhinoExampleCategory
B:
C: ${RHINO_EXAMPLE_CATEGORY}.iconBasedOn = type
D: ${RHINO_EXAMPLE_CATEGORY}.icon = com.sgi.rhexamp.ftr.Unknown
E: ${RHINO_EXAMPLE_CATEGORY}.icon.Clock = com.sgi.rhexamp.ftr.Clock
F: ${RHINO_EXAMPLE_CATEGORY}.icon.Printer = com.sgi.rhexamp.ftr.Printer
G: ${RHINO_EXAMPLE_CATEGORY}.icon.NetscapeExecutable = com.sgi.rhexamp.ftr.NetscapeE
```

With the resource defined as shown, then the icon displayed will be based on the "type" Attribute of the Item. For example, if the "type" is "Clock", then the FtrIcon `com.sgi.rhexamp.ftr.Clock` will be used. If the "type" Attribute is not one of "Clock", "Printer" or "NetscapeExecutable", then the `com.sgi.rhexamp.ftr.Unknown` will be displayed.

Another way to show different icons for each Item in the Category is to use the "iconModifiers" resource, defined as `<name>.iconModifiers` (see the [ICON_MODIFIERS](#) documentation for more information). This resource defines an array of Attributes of the Item that will be passed to the `set` method of `FtrIcon`. The `FtrIcon` can then use the Attributes to choose how to display the icon. The Attributes defined by the "iconModifiers" resource will be passed to the `FtrIcon` that is created. The Attributes will be passed to the `FtrIcon` both in the case where a default icon is used and in the case where a specific icon is used. These Attributes will be ignored if the icon is not an `FtrIcon`.

For example, if the `FtrIcons` for the `RhinoExampleCategory` could draw themselves differently based on the "mode" Attribute of the Item, then add the following to the resource file:

```
A: ${RHINO_EXAMPLE_CATEGORY}.iconModifiers0 = mode
```

In this case, the Item's "mode" Attribute will be passed to whatever `FtrIcon` is used (which - as described above - depends on the "type" Attribute).

Using a subclass of IconRenderer

To provide complete control of the icon that is used for an Item, it is also possible to subclass the `IconRenderer` class and provide the necessary Java code for rendering icons. See the [IconRenderer](#) documentation for more information. It is also possible to subclass the `ResourceBasedIconRenderer` to add to the existing functionality of the `ResourceBasedIconRenderer`.

How to use a Rhino NameRenderer

Table of Contents

- [Introduction](#)
 - [Before you begin](#)
 - [Specifying the NameRenderer to use for a Category](#)
 - [Using the ResourceBasedNameRenderer](#)
 - [Using a subclass of NameRenderer](#)
-

Introduction

This document is a reference for SGI software engineers who will be using NameRenderers for Rhino applications. NameRenderer is a class that can generate a name that represents a particular Item. Many of the Rhino infrastructure components use a NameRenderer to display the name associated with an Item, including the ItemView, the ItemTable, the ResultView, and the TreeView. The NameRenderer gives the programmer the ability to define what name an Item should have, and have that name used everywhere that the Item's name is displayed.

There is one NameRenderer for each Category. The NameRenderer is responsible for monitoring the Category and generating names for any Items that listeners have expressed interest in.

For more information on NameRenderers in general, see the [NameRenderer](#) documentation API documentation.

Before you begin

Before you begin to create a NameRenderer for a particular Category, you need to understand the names and terms that the Rhino infrastructure uses in relation to Categories. See the [The names of Categories on the client and on the server](#) documentation for more information.

Specifying the NameRenderer to use for a Category

The [HostContext](#) keeps track of which NameRenderer to use for each Category. To specify a Category's NameRenderer, place the `NAME_RENDERER` resource in the Category's resource file. For example, to specify that the RhinoExampleCategory should use the "com.sgi.rhexamp.category.NameRenderer" class as its NameRenderer, the following entry would be made

in /com/sgi/rhexamp/category/rhexampRhinoExampleCategoryP.properties:

```
com.sgi.rhexamp.category.rhexampRhinoExampleCategory.NameRenderer = com.sgi.rhexamp.
```

If this resource is not specified, then the HostContext creates a ResourceBasedNameRenderer object for the Category.

Using the ResourceBasedNameRenderer

The ResourceBasedNameRenderer is a subclass of NameRenderer that is used to display names if there is no NAME_RENDERER resource specified (see Specifying the NameRenderer to use for a Category). To use the ResourceBasedNameRenderer, you must place specific resources in the Category's resource file.

There are three resources to add to the Category's resource file:

1. The "categoryName" resource, defined as `<category name>.categoryName` (see the CATEGORY documentation for more information). This resource should contain a string that will be used as the name of the Category.
2. The "pluralCategoryName" resource, defined as `<category name>.pluralCategoryName` (see the CATEGORY_PLURAL documentation for more information). This resource should contain a string that will be used as the name of the Category in its plural form.
3. The "nameAttr" resource, defined as `<category name>.nameAttr` (see the NAME documentation for more information). This resource should contain the name of the Item's Attribute whose value will be used as the name of the Item.

As an example of these resource, the RhinoExample Category contains the following entries in its resource file:

```
{RHINO_EXAMPLE_CATEGORY}.categoryName = Rhino Example
{RHINO_EXAMPLE_CATEGORY}.pluralCategoryName = Rhino Examples
{RHINO_EXAMPLE_CATEGORY}.nameAttr = name
```

In this example, the name of the Category will be "Rhino Example", the plural name of the Category will be "Rhino Examples", and the name of the Item will be the contents of the "name" Attribute of the Item.

In addition to the three resources listed above, there are three more resources that control the way that the ResourceBasedNameRenderer works. These resources differ from the ones above in that they are not specific to a particular Category. It is expected that this resource will be common to all NameRenderers, and there are default values in the rhino tree. It is possible to override these resources in a product's `PackageP.properties` file, or individually in a specific Category's resource file.

1. The *ItemAndCategoryFormat* (see the ITEM_NAME_FORMAT documentation for more information). This resource gives the `FormatString` which will be used to combine the Category name and the Item name. `{0}` will be replaced by the Item name, `{1}` by the Category name. This is used in several UI components to display the name of the Item (for example: "Cluster c1").
2. The *ItemView.titleFormatString* (see the IV_TITLE_FORMAT documentation for more information). This resource gives the `FormatString` which will be used to combine the name of the Item, the name of the Category, and the name of the server into a single string. The name of the Item will be substituted in `{0}`, the name of the Category in `{1}`, and the name of the host that the GUI is connected to in `{2}`.
3. The *ItemTable.titleFormatString* (see the IT_TITLE_FORMAT documentation for more information). This resource gives the `FormatString` which will be used to combine the name of the


Category and the name of the server into a single string. The name of the Category will be substituted in {0} and the name of the host that the GUI is connected to in {1}.

For example, the default values for these properties, as defined in SysadmUIP.properties, are as follows:

```
A: ItemAndCategoryFormat = {1} {0}
B: ItemView.titleFormatString={1} {0} (on {2})
C: ItemTable.titleFormatString={0} (on {1})
```

Using a subclass of NameRenderer

To provide complete control of the name that is used for an Item, it is also possible to subclass the NameRenderer class and provide the necessary Java code for rendering names. See the [NameRenderer](#) documentation for more information. It is also possible to subclass the ResourceBasedNameRenderer to add to the existing functionality of the ResourceBasedNameRenderer.


[Web](#)
[Images](#)
[Groups](#)
[News](#)
[Froogle](#)
[New!](#)
[more »](#)

[Advanced Search](#)
[Preferences](#)

WebResults 1 - 10 of about 9,180 for **rhino group user interface**. (0.25 seconds)**RPM of Group User Interface/X**

... ecore-0.0.1-1, Enlightened Core X **interface** library, Linux/i386. ... sysadm_base-client-1.3.7-0, **Rhino** System Administration Client files, ...
fr.rpmfind.net/linux/RPM/User_Interface_X.html - 101k - [Cached](#) - [Similar pages](#)

Artificial Intelligence Research Group

... Computer Graphics, Computer Vision and Animation **group**. ... by Demonstration: Using AI techniques to improve **user interfaces**. ... Museum tour-guides: **Rhino** and Minerva ...
www.cs.washington.edu/ai/ - 42k - [Cached](#) - [Similar pages](#)

Rhino Architecture

... sysadmd sets its **user** id and its **group** id so ... This is the only way in which a **Rhino** client can ... Since TaskData (see **User Interface** section below) is derived from ...
oss.sgi.com/projects/rhino/tutorials/Architecture.html - 14k - [Cached](#) - [Similar pages](#)

Rhinoceros - What's new in 3.0

... By refreshing as needed, each **user** can see the ... The point cloud object type improves **Rhino's** performance when ... point cloud object can be used to **group** any number ...
www.rhino3d.com/3/whatisnew.htm - 30k - [Cached](#) - [Similar pages](#)

Rhinoceros - User comments

... with 16 MB RAM, and the **user interfaces** are excellent ... **Rhino** support is 10 times better than anything else I ... am truly impressed by the entire **group**." Jon Coign ...
www.rhino3d.com/comments.htm - 15k - [Cached](#) - [Similar pages](#)
[\[More results from www.rhino3d.com \]](#)

Luxor XML User Interface Language (XUL) Toolkit

... uses the Swing toolkit to render the **user interface**. ... Upstate New York Java Users **Group's** questions ... added using Netscape's 100 % Java **Rhino** JavaScript engine, ...
luxor-xul.sourceforge.net/ - 21k - [Cached](#) - [Similar pages](#)

Welcome To The Toronto CFUG

Where: **Rhino** - 1249 Queen St. W. ... Studio MX 2004 offers seamless integration and a common **user interface** across all tools to streamline web development and ...
www.cfugtoronto.org/ - 28k - Apr 14, 2004 - [Cached](#) - [Similar pages](#)

Scripting a user interface

... This support is provided using the **Rhino** JavaScript engine. ... Allows placement of an image as a **user interface** element. ... **group** box (<fieldset>) - id, title. ...
www.eclipse.org/documentation/html/plugins/org.eclipse.platform.doc.isv/doc/guide/scripting.htm - 15k - [Cached](#) - [Similar pages](#)

IRIS Project Plan

... responsible for communication with the **Rhino** board. The programmer will be responsible for the **user interface** and the database. The rest of the **group** will fill ...
www.cc.gatech.edu/classes/cs3302_98_spring/teams/team-b7/schedule3.html - 15k - [Cached](#) - [Similar pages](#)

Tivoli Distributed Monitoring (Advanced Edition): User's Guide ...

... Reboot; Adding the Notice **Group**; Installing Java ... Using Assistive Technologies; Keyboard Navigation of the **User Interface**; ... Appendix E. Use of **Rhino**: JavaScript for ...
publib.boulder.ibm.com/tividd/td/dmae/ SH19-4565-00/en_US/HTML/dmumst02.htm - 29k - [Cached](#) - [Similar pages](#)

Goooooooooooooogle ►

Result Page: **1** 2 3 4 5 6 7 8 9 10 **Next**

rhino group user interface

Search

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

[Google Home](#) - [Advertising Products](#) - [Business Solutions](#) - [About Google](#)

©2004 Google



UNITED STATES PATENT AND TRADEMARK OFFICE

[Home](#)
[Index](#)
[Search](#)
[System Alerts](#)
[eBusiness Center](#)
[News & Notices](#)
[Contact Us](#)

Trademark Electronic Search System(Tess)

TESS was last updated on Thu Apr 15 04:35:00 EDT 2004

[PTO HOME](#)
[TRADEMARK](#)
[TESS HOME](#)
[NEW USER](#)
[STRUCTURED](#)
[FREE FORM](#)
[BROWSE DICT](#)
[PREV LIST](#)
[NEXT LIST](#)
[BOTTOM](#)
[HELP](#)
[Logout](#)

Please logout when you are done to release system resources allocated for you.

[Start](#)
List
At:

OR

[Jump](#)
to
record:93 Records(s) found (This
page: 1 ~ 50)

(silicon graphics)[OW]

[Refine Search](#)

Current Search: S6: (silicon graphics)[OW] docs: 93 occ: 192

	Serial Number	Reg. Number	Word Mark	Check Status	Live/Dead
1	78229516		ALIAS SKETCHBOOK	TARR	LIVE
2	78273887			TARR	LIVE
3	78240435	2821359	TEZRO	TARR	LIVE
4	78365065		SHMEM	TARR	LIVE
5	78353290		SKETCHBOOK	TARR	LIVE
6	78182640	2803777	ALTIX	TARR	LIVE
7	78165491		VISO	TARR	LIVE
8	78143151	2709544	XFS	TARR	LIVE
9	76334536		THE DIGITAL CORKBOARD	TARR	LIVE
10	76165144		CREATE > WHAT'S > NEXT >	TARR	LIVE
11	76165143		PORTFOLIOWALL	TARR	LIVE
12	76160472	2759487	3DECEMBER.COM	TARR	LIVE
13	76160471	2759486	3DECEMBER	TARR	LIVE
14	76160470	2759485	3DEC	TARR	LIVE
15	76081740	2596959	CAN YOU IMAGINE	TARR	LIVE
16	76057130		Z-RAYS	TARR	DEAD
17	76184604		STUDIO TOOLS	TARR	DEAD
18	76195107		MEDIA COMMERCE	TARR	DEAD
19	76279589	2787974	IRIS FAILSAFE	TARR	LIVE
20	76350340		OPENMP	TARR	LIVE

21	76350213	2759937	SILICON GRAPHICS FUEL	TARR	LIVE
22	76279586	2743739	FAILSAFE	TARR	LIVE
23	76279576		LINUX FAILSAFE	TARR	DEAD
24	76184603		Z RAYS	TARR	DEAD
25	76110858		NUMAFLEX	TARR	LIVE
26	76110851		NUMALINK	TARR	LIVE
27	76060512		VISO	TARR	DEAD
28	76052045	2596870	ALIAS	TARR	LIVE
29	76037014	2576637	OPENML	TARR	LIVE
30	75214146	2168514	OCTANE	TARR	LIVE
31	75706196	2815943	REALITY CENTER	TARR	LIVE
32	75839877	2811898	DESIGN STUDIO	TARR	LIVE
33	75983314		REALITY CENTER	TARR	LIVE
34	75916455	2756592	OPENGL	TARR	LIVE
35	75941855	2705823	ALIAS WAVEFRONT	TARR	LIVE
36	75912147	2663825	UPSAFE	TARR	LIVE
37	75839878	2560531	DESIGNSTUDIO	TARR	LIVE
38	75702640	2579277	SGI	TARR	LIVE
39	75634597		COBALT	TARR	LIVE
40	75609926	2517956	SGI	TARR	LIVE
41	75574897	2517897	SGI	TARR	LIVE
42	75547245		SILICON GRAPHICS THE LEADER IN VISUAL COMPUTING	TARR	DEAD
43	75518307		SILICON GRAPHICS VISUAL	TARR	DEAD
44	75500559		VISUAL WORKSTATION	TARR	DEAD
45	75492571		VISUAL PC	TARR	DEAD
46	75472579		COSMO	TARR	DEAD
47	75465096		PROMIRA	TARR	DEAD
48	75340859		COBALT	TARR	DEAD
49	75212191	2127113	IRIX	TARR	LIVE
50	75173429	2220789	REALITY MONSTER	TARR	LIVE

[PTO HOME](#)
[TRADEMARK](#)
[TESS HOME](#)
[NEW USER](#)
[STRUCTURED](#)
[FREE FORM](#)
[Browse Dict](#)
[PREV LIST](#)
[NEXT LIST](#)
[TOP](#)

[HELP](#)

[HOME](#) |
 [INDEX](#) |
 [SEARCH](#) |
 [SYSTEM ALERTS](#) |
 [BUSINESS CENTER](#) |
 [NEWS&NOTICES](#) |
 [CONTACT US](#) |
 [PRIVACY STATEMENT](#)

Index	index by Group	index by Distribution	index by Vendor	index by creation date	index by Name	Mirrors	Search
-----------------------	--------------------------------	---------------------------------------	---------------------------------	--	-------------------------------	-------------------------	------------------------

RPM of Group User Interface/X

p 36

ami-1.2.2-4	Ami- a Korean Input Method System						
ami-1.2.2-3	Ami- a Korean Input Method System	linux/armv4l	linux/i386				
ami-1.0.11-13	Ami- a Korean Input Method System	linux/i386					
ami-1.0.11-9	Ami- a Korean Input Method System	linux/i386					
ami-1.0.11-4	Ami- a Korean Input Method System	linux/ppc	linux/ppc				
aterm-0.4.0-2	A color vt102 terminal emulator.	linux/ppc	Linux/i386	Linux/i386	linux/ppc	linux/ppc	
aterm-0.3.6-4	A color vt102 terminal emulator.	Linux/sparc	Linux/i386	Linux/alpha			
bbkeys-0.3.6-2a	a completely configurable key-combo grabber for blackbox	linux/ppc					
bbkeysconf-1.3-2a	GUI Configurator for bbkeys	linux/ppc					
bitmap-fonts-0.3-2	Selected set of bitmap fonts	linux/noarch	linux/noarch	linux/noarch			
bitmap-fonts-0.2-2	Selected set of bitmap fonts	linux/noarch					
bitstream-vera-fonts-1.10-2	Bitstream Vera Fonts						
doxygen-doxywizard-1.2.18-3	A GUI for creating and editing configuration files.	linux/armv4l	linux/i386	linux/i386			
doxygen-doxywizard-1.2.14-8	A GUI for creating and editing configuration files.	linux/i386					
doxygen-doxywizard-1.2.14-4	A GUI for creating and editing configuration files.	linux/i386					
doxygen-doxywizard-1.2.8.1-1	a GUI front-end for creating and editing configuration files	linux/ppc	linux/ppc				
ecore-0.0.2-1	Enlightened Core X	Linux/i386					

Index by Vendor p 31
pages 67

	interface library							
ecore-0.0.1-1	Enlightened Core X interface library	Linux/i386						
epplets-0.7-1	Enlightenment Epplets	linux/i386	linux/i386					
epplets-0.6- 0.20000731cvs1	Enlightenment Epplets	Linux/i386						
Eterm-0.9.2-1	Enlightened terminal emulator	linux/i386						
Eterm-0.9.1-3	Enlightened terminal emulator	linux/ppc	linux/i386					
Eterm-0.9-18	An xterm replacement for Enlightenment users.	linux/ppc	linux/ppc	linux/ppc	linux/ppc	linux/ppc	linux/ppc	linux/ppc
Eterm-0.9-7	An xterm replacement for Enlightenment users.	Linux/sparc	Linux/i386	Linux/alpha				
Eterm- backgrounds-0.9- 7	Backgrounds for Eterm	Linux/sparc	Linux/i386	Linux/alpha				
Eterm-0.9-4	Enlightened terminal emulator	Linux/i386						
Eterm-0.9-2	Enlightened terminal emulator	Linux/i386						
Eterm-0.8.10-4	An xterm replacement for Enlightenment users.	Linux/i386						
evas-0.5.0-1	Enlightened Canvas Library	Linux/i386						
evas-0.0.4-1	Enlightened Canvas Library	Linux/i386						
evas-0.0.3-1	Enlightened Canvas Library	Linux/i386						
evas-0.0.2-1	Enlightened Canvas Library	Linux/i386						
evas-0.0.1-1	Enlightened Canvas Library	Linux/i386						
fonts-ja-8.0-9	Japanese bitmap fonts							
fonts-ja-8.0-7	Japanese bitmap fonts	linux/noarch	linux/noarch					
fonts-ja-8.0-4	Japanese bitmap fonts	linux/noarch						
fonts-ja-7.x-1	Japanese bitmap fonts	linux/noarch						
fonts-arabic-1.4- 1.1	Fonts for Arabic							
fonts-CP1251- 1.0-2asp	Russian and Ukrainian language fonts for the X Window System.	linux/noarch						

fonts-hebrew-0.80-1	Fonts for Hebrew			
fonts-hebrew-0.71-2	Fonts for Hebrew	linux/noarch	linux/noarch	linux/noarch
fonts-CP1251-75dpi-1.0-2asp	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch		
fonts-ISO10646-1-0.4-0.3asp	Unicode bitmap fonts for XFree86 with cyrillic glyphs	linux/noarch		
fonts-ISO8859-2-1.0-11	Central European language fonts for the X Window System.	linux/noarch	linux/noarch	linux/noarch
fonts-ISO8859-2-1.0-8	Central European language fonts for the X Window System.	linux/noarch		
fonts-ISO8859-2-1.0-4	Central European language fonts for the X Window System.	linux/noarch		
fonts-ISO8859-5-1.0-2asp	Russian and Ukrainian language fonts for the X Window System.	linux/noarch		
fonts-ISO8859-7-1.0-2	Greek language fonts for the X Window System.	linux/noarch		
fonts-ISO8859-2-100dpi-1.0-11	A set of 100dpi Central European language fonts for X.	linux/noarch	linux/noarch	linux/noarch
fonts-ISO8859-2-75dpi-1.0-11	A set of 75dpi Central European language fonts for X.	linux/noarch	linux/noarch	linux/noarch
fonts-ISO8859-2-100dpi-1.0-8	A set of 100dpi Central European language fonts for X.	linux/noarch		
fonts-ISO8859-2-75dpi-1.0-8	A set of 75dpi Central European language fonts for X.	linux/noarch		
fonts-ISO8859-2-100dpi-1.0-4	A set of 100dpi Central European language fonts for X.	linux/noarch		
fonts-ISO8859-2-75dpi-1.0-4	A set of 75dpi Central European language fonts for X.	linux/noarch		
fonts-ISO8859-5-75dpi-1.0-2asp	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch		
fonts-ISO8859-7-100dpi-1.0-2	ISO 8859-7 fonts in 100 dpi resolution for the X Window System.	linux/noarch		
fonts-ISO8859-7-75dpi-1.0-2	ISO 8859-7 fonts in 75 dpi resolution for the X Window System.	linux/noarch		
fonts-ISO8859-7-Type1-1.0-2	Type 1 scalable Greek (ISO 8859-7) fonts	linux/noarch		
fonts-KOI8-R-1.0-5.1asp	Russian and Ukrainian language fonts for the X Window System.	linux/noarch		
fonts-KOI8-R-1.0-5	Russian and Ukrainian language fonts for the X Window System.	linux/noarch	linux/noarch	
fonts-KOI8-R-1.0-3	Russian and Ukrainian language fonts for the X Window System.		linux/noarch	

fonts-KOI8-R-1.0-1	Russian and Ukrainian language fonts for the X Window System.	linux/noarch	
fonts-KOI8-R-100dpi-1.0-5.1asp	KOI8-R fonts in 100 dpi resolution for the X Window System.	linux/noarch	
fonts-KOI8-R-75dpi-1.0-5.1asp	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch	
fonts-KOI8-R-100dpi-1.0-5	KOI8-R fonts in 100 dpi resolution for the X Window System.	linux/noarch	linux/noarch
fonts-KOI8-R-75dpi-1.0-5	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch	linux/noarch
fonts-KOI8-R-100dpi-1.0-3	KOI8-R fonts in 100 dpi resolution for the X Window System.	linux/noarch	
fonts-KOI8-R-75dpi-1.0-3	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch	
fonts-KOI8-R-100dpi-1.0-1	KOI8-R fonts in 100 dpi resolution for the X Window System.	linux/noarch	
fonts-KOI8-R-75dpi-1.0-1	A set of 75 dpi Russian and Ukrainian language fonts for X.	linux/noarch	
freebanglafont-0.5-1	A set of Unicode compliant Bangla fonts under the GNU GPL	linux/noarch	
gdm-2.4.4.5-1asp	The GNOME Display Manager.	linux/i386	
gdm-2.4.4.5-1	The GNOME Display Manager.		
gdm-2.4.1.3-5.2asp	The GNOME Display Manager.	linux/i386	
gdm-2.4.1.3-5.1asp	The GNOME Display Manager.	linux/i386	
gdm-2.4.1.3-5.1	The GNOME Display Manager.	linux/i386	
gdm-2.4.1.3-5	The GNOME Display Manager.	linux/armv4l	linux/i386
gdm-2.4.0.7-14	The GNOME Display Manager.	linux/i386	
gdm-2.4.0.7-13	The GNOME Display Manager.	linux/i386	
gdm-2.2.5.5-1	The GNOME Display Manager.	linux/i386	
gdm-2.2.5.4-1	The GNOME Display Manager.	linux/i386	linux/alpha
gdm-2.2.5.3-1	The GNOME Display Manager.	linux/alpha	
gdm-2.2.5.2-1	The GNOME Display Manager.	linux/i386	linux/alpha
gdm-2.2.5.1-1	The GNOME Display Manager.	linux/i386	linux/alpha
gdm-2.2.4.3-1	The GNOME Display Manager.	linux/i386	linux/alpha
gdm-2.2.4.2-1	The GNOME Display Manager.	linux/i386	linux/alpha
gdm-2.2.4.1-1	The GNOME Display Manager.	linux/i386	
gdm-2.2.3.2-1	The GNOME Display Manager.	linux/i386	
gdm-2.2.3.1-23asp	The GNOME Display Manager.		
gdm-2.2.3.1-23	The GNOME Display Manager.	linux/i386	
gdm-2.2.3.1-22	The GNOME Display Manager.	linux/i386	
gdm-2.2.3.1-21asp	The GNOME Display Manager.	linux/i386	
gdm-2.2.3.1-21	The GNOME Display Manager.	linux/ia64	linux/i386
gdm-2.2.3.1-20.1	The GNOME Display Manager.		

4/15/04

6	Display Manager.			
gdm- 1.0.0-36	GNOME Display Manager.			
gdm- 1.0.0- 12rh	GNOME Display Manager.	Linux/sparc	Linux/alpha	Linux/i386
gdm- 1.0.0- 11rh	GNOME Display Manager.			
glms- 1.03-14	A GNOME hardware monitoring applet.	linux/i386		
glms- 1.03-11	A GNOME hardware monitoring applet.			
gnome- kerberos- 0.3.2-1	Kerberos 5 tools for GNOME.			
gnome- kerberos- 0.3.1-9	Kerberos 5 tools for GNOME.	linux/armv4l	linux/i386	linux/i386
gnome- kerberos- 0.3.1-7	Kerberos 5 tools for GNOME.	linux/i386		
gnome- kerberos- 0.3-4	Kerberos 5 tools for GNOME.	linux/i386		
gnome- kerberos- 0.2.2-4	Kerberos 5 tools for GNOME.	linux/ppc	linux/ppc	
gnome-kerberos- 0.2.1-1	Kerberos 5 tools for GNOME.		linux/ppc	
gnome-kerberos- 0.2-1	Kerberos 5 tools for GNOME.	Linux/i386	Linux/sparc	Linux/alpha
gqview-1.3.5- 1.rh9	An image viewer.	linux/i386		
gqview-1.3.5-1asp	An image viewer.	linux/i386		
gqview-1.3.4- 1.rh9	An image viewer.	linux/i386		
gqview-1.3.3- 1.rh9	An image viewer.	linux/i386		

gqview-1.3.2-1.rh9	An image viewer.	linux/i386			
gqview-1.3.1-1.rh8	An image viewer.	linux/i386			
gqview-1.3.0-2.rh8	An image viewer.	linux/i386			
gqview-1.3.0-1.rh8	An image viewer.	linux/i386			
gqview-1.2.2-1.rh8	An image viewer.	linux/i386			
gqview-1.2.1-6	An image viewer.				
gqview-1.2.1-3	An image viewer.	linux/armv4l	linux/i386	linux/i386	
gqview-1.2.1-1.rh8	An image viewer.	linux/i386			
gqview-1.0.2-2	An image viewer.	linux/i386			
gqview-0.8.1-5	An image viewer.	linux/ppc	linux/ppc	linux/i386	
gqview-0.8.1-2	An image viewer.	linux/ppc			
gqview-0.7.0-4	An image viewer.	Linux/i386	Linux/ppc	Linux/sparc	Linux/alpha
gqview-0.7.0-2	graphics file browser utility	Linux/armv4l			
gqview-0.7.0-1	graphics file browser utility	Linux/i386	Linux/i386	Linux/alpha	Linux/sparc
gqview-0.6.0-2	graphics file browser utility		Linux/sparc	Linux/alpha	Linux/i386
gthumb-2.0.2-1	Image viewer, editor, organizer				
gthumb-2.0.1-1	Image viewer, editor, organizer	linux/armv4l	linux/i386	linux/i386	
hanterm-xf-p19-15	Hangul Terminal for X Window System.	linux/ppc	linux/i386		
hanterm-xf-2.0.5-5.AS21.1	Hangul Terminal for X Window System.				
hanterm-xf-2.0.5-5.8.0	Hangul Terminal for X Window System.	linux/i386			
hanterm-xf-2.0.5-5.7.4	Hangul Terminal for X Window System.	linux/i386	linux/i386	linux/ia64	
hanterm-xf-2.0.5-5	Hangul Terminal for X Window System.	linux/armv4l	linux/i386		
hanterm-xf-2.0.0-6	Hangul Terminal for X Window System.	linux/i386			
hunglish-X11-1.12-1	Dual English/Hungarian keyboard layout for X	linux/noarch			
im-sdk-20030118-5	IIIMF Input Method Software Developpers Kit	linux/armv4l	linux/i386		
im-sdk-20030118-3	IIIMF Input Method Software Developpers Kit	linux/i386			
im-sdk-20020813-1	IIIMF	linux/i386			
im-sdk-20011223-3	IIIMF	linux/i386			
im-sdk-11_4-1	IIIMF Input Method Software				

Develloppers Kit

imwheel-1.0.0pre4-1	A utility to make wheel mice work under X	linux/i386			
jisksp14-0.1-11	14 dots jis auxiliary kanji font	linux/noarch	linux/noarch		
jisksp14-0.1-9	14 dots jis auxiliary kanji font	linux/noarch			
jisksp14-0.1-7	14 dots jis auxiliary kanji font	linux/noarch			
jisksp14-0.1-6	14 dots jis auxiliary kanji font	linux/noarch	linux/noarch		
jisksp16-1990-0.1-11	16 dots jis auxiliary kanji font	linux/noarch	linux/noarch		
jisksp16-1990-0.1-9	16 dots jis auxiliary kanji font	linux/noarch			
jisksp16-1990-0.1-6	16 dots jis auxiliary kanji font	linux/noarch	linux/noarch	linux/noarch	
kappa20-0.3-12	Kappa 20dot Font	linux/noarch	linux/noarch		
kappa20-0.3-10	Kappa 20dot Font	linux/noarch			
kappa20-0.3-7	Kappa 20dot Font	linux/noarch	linux/noarch	linux/noarch	
kinput2-v3.1-11	kinput2 - kanji input server for X11				
kinput2-v3.1-4	kinput2 - kanji input server for X11				
kinput2-v3-18	kinput2 - kanji input server for X11				
kinput2-v3-14	kinput2 - kanji input server for X11				
kinput2-canna-wnn6-v3.1-4	kinput2 for both Canna and Wnn6	linux/i386	linux/armv4l		
kinput2-canna-wnn6-v3-18	kinput2 for both Canna and Wnn6	linux/i386			
kinput2-canna-wnn6-v3-14	kinput2 for both Canna and Wnn6	linux/i386			
knm_new-1.1-11	Kaname-cho font, revised version				
knm_new-1.1-10	Kaname-cho font, revised version	linux/noarch	linux/noarch		
knm_new-1.1-8	Kaname-cho font, revised version	linux/noarch			
knm_new-1.1-5	Kaname-cho font, revised version	linux/noarch	linux/noarch	linux/noarch	linux/noarch
kterm-6.2.0-37	A Kanji (Japanese character set) terminal emulator for X.				
kterm-6.2.0-36	A Kanji (Japanese character set) terminal emulator for X.	linux/armv4l	linux/i386		
	A Kanji (Japanese				

kterm-6.2.0-30	character set) terminal emulator for X.	linux/i386			
kterm-6.2.0-28	A Kanji (Japanese character set) terminal emulator for X.	linux/i386			
kterm-6.2.0-22	A Kanji (Japanese character set) terminal emulator for X.	linux/ppc	linux/ppc		
kterm-6.2.0-14	A Kanji (Japanese character set) terminal emulator for X.	linux/ppc	linux/ppc		
kterm-6.2.0-10	A Kanji (Japanese character set) terminal emulator for X.	Linux/i386	Linux/ppc	Linux/sparc	Linux/alpha
kterm-6.2.0-8	A Kanji (Japanese character set) terminal emulator for X.	Linux/armv4l			
lesstif-clients-0.89.9-2.1	UIL and xmbind, two separate LessTif add-ons.	Linux/ppc			
lesstif-clients-0.89.9-2	UIL and xmbind, two separate LessTif add-ons.	Linux/sparc	Linux/i386	Linux/alpha	
libtabe-0.2.6-6	Chinese lexicons library for xcin-2.5.3				
libtabe-0.2.6-3	Chinese lexicons library for xcin-2.5.3	linux/armv4l	linux/i386		
libtabe-0.2.6-1	Chinese lexicons library for xcin-2.5.3	linux/i386			
libtabe-0.2.4a-10	Chinese lexicons library for xcin-2.5.3	linux/i386			
libtabe-0.2-6	Chinese lexicons library for xcin-2.5.2	linux/ppc	linux/ppc		
miniChinput-0.0.3-53	A Chinese XIM server.				
miniChinput-0.0.3-37	A Chinese XIM server.	linux/armv4l	linux/i386		
miniChinput-0.0.3-27	A Chinese XIM server.	linux/i386			
miniChinput-0.0.3-20	A Chinese XIM server.	linux/i386			
miniChinput-0.0.3-18	A Chinese XIM server.	linux/i386			
miniChinput-0.0.2-1	A Chinese XIM server	linux/ppc	linux/ppc		
mkfontalias-20000521-3.9	creates a fonts.alias file from the fonts.dir file	linux/noarch			
moonclock-1.0-15	A clock which also displays the current moon	Linux/sparc	Linux/i386	Linux/alpha	

	phase.						
morseall-0.1.7-1	Morseall - morse code controlled shell	Linux/i386					
msttcorefonts-1.3-fr3	TrueType core fonts for the web.						
msttcorefonts-1-1	TrueType core fonts for the web						
OffiX-Clipboard-2.4-9	A drag and drop clipboard patch for xclipboard.	Linux/sparc	Linux/i386	Linux/alpha			
olvwm-4.2n-1	OpenLook Virtual Window Manager.	Linux/i386					
rxvt-2.7.8-4	A color VT102 terminal emulator for the X Window System.	linux/i386	linux/i386	linux/i386	linux/ia64	linux/i386	
rxvt-2.7.8-3.7.1.1	A color VT102 terminal emulator for the X Window System.	linux/i386	linux/i386				
rxvt-2.7.8-3.7.0.1	A color VT102 terminal emulator for the X Window System.	linux/i386					
rxvt-2.7.8-3.6.2.1	A color VT102 terminal emulator for the X Window System.	linux/i386					
rxvt-2.7.8-3asp	A color VT102 terminal emulator for the X Window System.	linux/i386					
rxvt-2.7.8-3	A color VT102 terminal emulator for the X Window System.	linux/i386					
rxvt-2.7.6-4	A color VT102 terminal emulator for the X Window System.	linux/ppc	linux/ppc				
rxvt-2.7.3-2	rxvt - terminal emulator in an X window	Linux/i386	Linux/i386	Linux/i386			
rxvt-2.6.3-2	A color VT102 terminal emulator for the X Window System.	linux/ppc					

rxvt-2.6.3-1	rxvt - terminal emulator in an X window	Linux/i386	Linux/i386	
rxvt-2.6.1-8	A color VT102 terminal emulator for the X Window System.	Linux/i386	Linux/sparc	Linux/alpha
rxvt-2.6.1-6	A color VT102 terminal emulator for the X Window System.	Linux/ppc		
rxvt-2.6.1-1	A color VT102 terminal emulator for the X Window System.	Linux/armv4l		
rxvt-2.6.0-2	A color VT102 terminal emulator for the X Window System.	Linux/i386	Linux/alpha	Linux/sparc
skkinput-2.06.3-4	SKK like Japanese-input application			
skkinput-2.06.3-3	SKK like Japanese-input application		linux/armv4l	linux/i386
skkinput-2.04-2	SKK like Japanese-input application		linux/i386	
skkinput-2.03-9	SKK like Japanese-input application		linux/i386	
skkinput-2.03-7	SKK like Japanese-input application		linux/ppc	linux/ppc
sysadm_base-client-1.3.7-0	Rhino System Administration Client files	Linux/i386		
taipeifonts-1.2-23	Taipei Chinese Big 5 Fonts	linux/noarch	linux/noarch	
taipeifonts-1.2-20	Taipei Chinese Big 5 Fonts	linux/noarch		
taipeifonts-1.2-16	Taipei Chinese Big 5 Fonts	linux/noarch		
taipeifonts-1.2-15	Taipei Chinese Big 5 Fonts	linux/noarch	linux/noarch	
tipa-type1-X11-3.1-1	IPA Type1 fonts for X11	linux/noarch		
TrueTypeFonts_jp-2-5	Free Japanese True Type Font	linux/noarch	linux/noarch	
tsclient-0.120-1	Client for VNC and Windows Terminal Server			
tsclient-0.104-1	Client for VNC and Windows Terminal Server	linux/armv4l	linux/i386	linux/i386
ttfonts-zh_CN-2.13-0	TrueType Fonts -- GB ming and kai face and Zhongyi 18030 Song face.			
ttfonts-zh_CN-2.12-1	TrueType Fonts -- GB ming and kai face and Zhongyi 18030 Song	linux/noarch	linux/noarch	

	face.			
ttfonts-zh_CN-2.11-29	TrueType Fonts -- GB ming and kai face and Zhongyi 18030 Song face.	linux/noarch		
ttfonts-zh_TW-2.11-22	Arphic TrueType Font -- Big5 ming and kai face.			
ttfonts-zh_CN-2.11-21	TrueType Fonts -- GB ming and kai face and Zhongyi 18030 Song face.	linux/noarch		
ttfonts-zh_TW-2.11-19	Arphic TrueType Font -- Big5 ming and kai face.	linux/noarch	linux/noarch	
ttfonts-zh_CN-2.11-17	Arphic TrueType Font -- GB ming and kai face.	linux/noarch	linux/noarch	
ttfonts-zh_TW-2.11-15	Arphic TrueType Font -- Big5 ming and kai face.	linux/noarch		
ttfonts-zh_TW-2.11-10	Arphic TrueType Font -- Big5 ming and kai face.	linux/noarch		
ttfonts-zh_CN-2.11-5	Arphic TrueType Font -- GB ming and kai face.			
ttfonts-zh_TW-2.11-5	Arphic TrueType Font -- Big5 ming and kai face.	linux/noarch	linux/noarch	
ttfonts-ja-1.2-29	Free Japanese TrueType fonts			
ttfonts-ja-1.2-21	Free Japanese TrueType fonts	linux/noarch	linux/noarch	
ttfonts-ja-1.2-16	Free Japanese TrueType fonts	linux/noarch		
ttfonts-ja-1.2-8	Free Japanese TrueType fonts	linux/noarch		
ttfonts-ko-1.0.11-29	Baekmuk Korean TrueType Fonts			
ttfonts-ko-1.0.11-26	Baekmuk Korean TrueType Fonts	linux/noarch	linux/noarch	
ttfonts-ko-1.0.11-21	Baekmuk Korean TrueType fonts.	linux/noarch		
ttfonts-ko-1.0.11-13	Baekmuk Korean TrueType fonts.	linux/noarch	linux/noarch	
ttfonts-ko-1.0.11-9	Baekmuk Korean TrueType fonts.	linux/noarch		
ttfonts-ko-1.0-11	Baekmuk Korean TrueType fonts.	linux/noarch		
ttfonts-1.0-9	Some TrueType fonts	linux/noarch		
ttfonts-ja-1.0-8	Free Japanese TrueType fonts			
ttfonts-ja-1.0-7	Free Japanese TrueType fonts	linux/noarch	linux/noarch	linux/noarch
ttfonts-1.0-4	Some TrueType fonts	linux/noarch	linux/noarch	linux/noarch
unifont-1999.04.30-2	Unicode Font for X	Linux/noarch		
urw-fonts-2.1-5.1	Free versions of the 35 standard PostScript fonts.			
urw-fonts-2.0-29asp	Free versions of the 35 standard PostScript fonts.	linux/noarch		
urw-fonts-2.0-29	Free versions of the 35 standard PostScript fonts.	linux/noarch	linux/noarch	

urw-fonts-2.0-26	Free versions of the 35 standard PostScript fonts.	linux/noarch			
urw-fonts-2.0-17	Free versions of the 35 standard PostScript fonts.	linux/noarch			
urw-fonts-2.0-12	Free versions of the 35 standard PostScript fonts.	linux/noarch	linux/noarch	linux/noarch	
urw-fonts-2.0-8	Free versions of the 35 standard PostScript fonts.	linux/noarch			
urw-fonts-2.0-4	Free versions of the 35 standard PostScript fonts.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
urw-fonts-1.1-ximian.1	Fonts for gnome-print	solaris2.7/sparc	solaris2.7/sparc	solaris2.7/sparc	solaris2.7/sparc
urw-fonts-1.1-9	Free versions of the 35 standard PostScript fonts.	Linux/noarch			
urw-fonts-1.1-8	Free versions of the 35 standard PostScript fonts.	Linux/i386	Linux/noarch	Linux/noarch	Linux/i386
vgafonts-1.0-2	X ANSI Fonts for DOSEMU, Terminals, Etc.				
vnc-3.3.6-3	The original open-source cross-platform remote control solution	linux/i386			
vnc-server-3.3.6-3	A VNC server.	linux/i386			
vnc-server-3.3.3r2+tight1.2.6-1	A VNC server.	linux/ppc			
vnc-server-3.3.3r2+tight1.2.0-1	A VNC server.	linux/i386			
vnc-server-3.3.3r2-47	A VNC server.	linux/i386	linux/i386		
vnc-server-3.3.3r2-39.2	A VNC server.	linux/i386			
vnc-server-3.3.3r2-39	A VNC server.	linux/i386			
vnc-server-3.3.3r2-30a	A VNC server.	linux/ppc			
vnc-server-3.3.3r2-28.2	A VNC server.	linux/i386	linux/i386		

vnc-server-3.3.3r2-28	A VNC server.	linux/i386				
vnc-server-3.3.3r2-18.6	A VNC server.	linux/i386	linux/i386	linux/i386	linux/i386	linux/i386
vnc-server-3.3.3r2-18.4	A VNC server.	linux/alpha	linux/alpha			
vnc-server-3.3.3r2-18a	A VNC server.	linux/ppc	linux/ppc			
vte-0.11.10-4	An experimental terminal emulator.					
vte-0.11.10-2	An experimental terminal emulator.	linux/i386				
vte-0.10.25-1	An experimental terminal emulator.	linux/armv4l	linux/i386	linux/i386		
vte-0.8.19-2	An experimental terminal emulator.	linux/i386				
vte-0.8.19-1	An experimental terminal emulator.	linux/i386				
waseda-X11-fonts-920515-8	waseda multilingual fonts for X11	linux/noarch				
wdm-1.20-1	WINGs Display Manager.	Linux/i386				
wdm-1.16-1	WINGs Display Manager.	Linux/i386				
wmapm-1.1-2	dockable clock applet for Window Maker	linux/i386	linux/i386			
wmapm-1.1-1	dockable clock applet for Window Maker	linux/i386	linux/i386			
wmclock-1.0.12.2-3	dockable clock applet for Window Maker	linux/i386				
wmclock-1.0.12.2-2	dockable clock applet for Window Maker	linux/i386	linux/i386			
wmix-3.0-4	Dockapp sound mixer for OSS or ALSA	linux/i386				
wmix-3.0-2	dockable clock applet for Window Maker	linux/i386	linux/i386			
wmppp-1.3.0-1	PPP dial control and network load monitor with NeXTStep look	linux/i386				
X11R6-contrib-3.3.6-3	A collection of user-contributed X Window System programs	Linux/ppc				
X11R6-contrib-3.3.2-11	A collection of user-contributed X Window System programs.	Linux/sparc	Linux/i386	Linux/alpha		
	A collection of user-					

X11R6-contrib-3.3.2-6	contributed X Window System programs.	Linux/armv4l				
x2vnc-1.2-1	Control a linux and a windows box using two monitors, one keyboard, and one mouse.	Linux/i386				
x3270-3.1.1.9-6	An X Window System based IBM 3278/3279 terminal emulator.	linux/ppc	linux/ppc			
x3270-3.1.1.9-3	An X Window System based IBM 3278/3279 terminal emulator.	Linux/alpha	Linux/ppc	Linux/sparc	Linux/i386	
x3270-3.1.1.6-8	An X Window System based IBM 3278/3279 terminal emulator.	Linux/armv4l				
xalf-0.11-5	A utility to provide feedback when starting X11 applications.	linux/i386				
xalf-0.11-4	A utility to provide feedback when starting X11 applications.	linux/ppc	linux/ppc			
xawtv-tv-fonts-3.90-1asp	Bitmap fonts for xawtv	linux/i386				
xawtv-tv-fonts-3.85-1.1asp	Bitmap fonts for xawtv	linux/i386				
xawtv-tv-fonts-3.81-6	Bitmap fonts for xawtv	linux/i386	linux/armv4l			
xcin-2.5.3.pre3-17	An X Input Method Server for Chinese.					
xcin-2.5.3.pre3-11	An X Input Method Server for Chinese.	linux/armv4l	linux/i386			
xcin-2.5.3.pre3-6	An X Input Method Server for Chinese.	linux/i386				
xcin-2.5.3.pre2-10	An X Input Method Server for Chinese.	linux/i386				
xcin-2.5.2.3-6	An X Input Method Server for Chinese.					
XFree86-4.3.0-42	The basic fonts, programs and docs for an X workstation.					
XFree86-4.3.0-2.90.55	The basic fonts, programs and docs for an X workstation.	linux/i386				
XFree86-sdk-4.3.0-2.90.55	XFree86 SDK for X server driver module development	linux/i386				
XFree86-tools-4.3.0-2.90.55	Various tools for XFree86	linux/i386				

XFree86-twm-4.3.0-2.90.55	A simple window manager	linux/i386	
XFree86-xauth-4.3.0-2.90.55	X authority file utility	linux/i386	
XFree86-xdm-4.3.0-2.90.55	X Display Manager	linux/i386	
XFree86-Xnest-4.3.0-2.90.55	A nested XFree86 server.	linux/i386	
XFree86-Xvfb-4.3.0-2.90.55	A virtual framebuffer X Windows System server for XFree86.	linux/i386	
XFree86-4.3.0-2.90.43	The basic fonts, programs and docs for an X workstation.	linux/i386	linux/i386
XFree86-sdk-4.3.0-2.90.43	XFree86 SDK for X server driver module development	linux/i386	linux/i386
XFree86-tools-4.3.0-2.90.43	Various tools for XFree86	linux/i386	linux/i386
XFree86-twm-4.3.0-2.90.43	A simple window manager	linux/i386	linux/i386
XFree86-xauth-4.3.0-2.90.43	X authority file utility	linux/i386	linux/i386
XFree86-xdm-4.3.0-2.90.43	X Display Manager	linux/i386	linux/i386
XFree86-Xnest-4.3.0-2.90.43	A nested XFree86 server.	linux/i386	linux/i386
XFree86-Xvfb-4.3.0-2.90.43	A virtual framebuffer X Windows System server for XFree86.	linux/i386	linux/i386
XFree86-4.3.0-2_nw3	The basic fonts, programs and docs for an X workstation.	linux/armv4l	
XFree86-tools-4.3.0-2_nw3	Various tools for XFree86	linux/armv4l	
XFree86-twm-4.3.0-2_nw3	A simple window manager	linux/armv4l	
XFree86-xauth-4.3.0-2_nw3	X authority file utility	linux/armv4l	
XFree86-xdm-4.3.0-2_nw3	X Display Manager	linux/armv4l	
XFree86-Xnest-4.3.0-2_nw3	A nested XFree86 server.	linux/armv4l	
XFree86-Xvfb-4.3.0-2_nw3	A virtual framebuffer X Windows System server for XFree86.	linux/armv4l	
XFree86-4.3.0-2	The basic fonts, programs and docs for an X workstation.	linux/i386	linux/i386
XFree86-tools-4.3.0-2	Various tools for XFree86	linux/i386	linux/i386
XFree86-twm-4.3.0-2	A simple window manager	linux/i386	linux/i386
XFree86-xauth-4.3.0-2	X authority file utility	linux/i386	linux/i386

XFree86-xdm-4.3.0-2	X Display Manager	linux/i386	linux/i386
XFree86-Xnest-4.3.0-2	A nested XFree86 server.	linux/i386	linux/i386
XFree86-Xvfb-4.3.0-2	A virtual framebuffer X Windows System server for XFree86.	linux/i386	linux/i386
XFree86-4.2.1-23	The basic fonts, programs and docs for an X workstation.	linux/i386	
XFree86-tools-4.2.1-23	Various tools for XFree86	linux/i386	
XFree86-twm-4.2.1-23	A simple window manager	linux/i386	
XFree86-xauth-4.2.1-23	X authority file utility	linux/i386	
XFree86-xdm-4.2.1-23	X Display Manager	linux/i386	
XFree86-Xnest-4.2.1-23	A nested XFree86 server.	linux/i386	
XFree86-Xvfb-4.2.1-23	A virtual framebuffer X Windows System server for XFree86.	linux/i386	
XFree86-4.2.1-13.73.23	The basic fonts, programs and docs for an X workstation.	linux/i386	
XFree86-tools-4.2.1-13.73.23	Various tools for XFree86	linux/i386	
XFree86-twm-4.2.1-13.73.23	A simple window manager	linux/i386	
XFree86-xdm-4.2.1-13.73.23	X Display Manager	linux/i386	
XFree86-xf86cfg-4.2.1-13.73.23	XFree86 configuration program	linux/i386	
XFree86-Xnest-4.2.1-13.73.23	A nested XFree86 server.	linux/i386	
XFree86-Xvfb-4.2.1-13.73.23	A virtual framebuffer X Windows System server for XFree86.		linux/i386
XFree86-4.2.0-72	The basic fonts, programs and docs for an X workstation.		linux/i386
XFree86-tools-4.2.0-72	Various tools for XFree86		linux/i386
XFree86-twm-4.2.0-72	A simple window manager		linux/i386
XFree86-xauth-4.2.0-72	X authority file utility		linux/i386
XFree86-xdm-4.2.0-72	X Display Manager		linux/i386
XFree86-Xnest-4.2.0-72	A nested XFree86 server.		linux/i386
XFree86-Xvfb-4.2.0-72	A virtual framebuffer X Windows System server for XFree86.		linux/i386
XFree86-4.2.0-8.6asp	The basic fonts, programs and docs for an X workstation.		linux/i386
XFree86-tools-4.2.0-8.6asp	Various tools for XFree86		linux/i386
XFree86-twm-4.2.0-8.6asp	A simple window manager		linux/i386
XFree86-xdm-4.2.0-8.6asp	X Display Manager		linux/i386
XFree86-xf86cfg-4.2.0-8.6asp	XFree86 configuration program		linux/i386
XFree86-4.2.0-8	The basic fonts, programs and docs for an X workstation.		linux/i386

XFree86-tools-4.2.0-8	Various tools for XFree86	linux/i386		
XFree86-twm-4.2.0-8	A simple window manager	linux/i386		
XFree86-xdm-4.2.0-8	X Display Manager	linux/i386		
XFree86-xf86cfg-4.2.0-8	XFree86 configuration program	linux/i386		
XFree86-4.2.0-6.30e	The basic fonts, programs and docs for an X workstation.	linux/ppc		
XFree86-tools-4.2.0-6.30e	Various tools for XFree86	linux/ppc		
XFree86-twm-4.2.0-6.30e	A simple window manager	linux/ppc		
XFree86-xdm-4.2.0-6.30e	X Display Manager	linux/ppc		
XFree86-xf86cfg-4.2.0-6.30e	XFree86 configuration program	linux/ppc		
XFree86-4.2.0-6.30d	The basic fonts, programs and docs for an X workstation.	linux/ppc		
XFree86-tools-4.2.0-6.30d	Various tools for XFree86	linux/ppc		
XFree86-twm-4.2.0-6.30d	A simple window manager	linux/ppc		
XFree86-xdm-4.2.0-6.30d	X Display Manager	linux/ppc		
XFree86-xf86cfg-4.2.0-6.30d	XFree86 configuration program	linux/ppc		
XFree86-4.1.0-56.EL	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-50.EL	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-50	The basic fonts, programs and docs for an X workstation.	linux/i386	linux/i386	linux/ia64
XFree86-tools-4.1.0-50	Various tools for XFree86	linux/ia64	linux/i386	linux/i386
XFree86-twm-4.1.0-50	A simple window manager	linux/i386	linux/i386	linux/ia64
XFree86-xdm-4.1.0-50	X Display Manager	linux/i386	linux/i386	linux/ia64
XFree86-xf86cfg-4.1.0-50	XFree86 configuration program	linux/i386	linux/i386	
XFree86-4.1.0-49.RHEL	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-46	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-44	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-29	The basic fonts, programs and docs for an X workstation.			
XFree86-4.1.0-25	The basic fonts, programs and docs for an X workstation.	linux/ia64	linux/alpha	

XFree86-tools-4.1.0-25	Various tools for XFree86	linux/alpha	linux/ia64
XFree86-twm-4.1.0-25	A simple window manager	linux/alpha	linux/ia64
XFree86-xdm-4.1.0-25	X Display Manager	linux/alpha	linux/ia64
XFree86-xf86cfg-4.1.0-25	XFree86 configuration program	linux/alpha	
XFree86-4.1.0-15.asp	The basic fonts, programs and docs for an X workstation.	linux/i386	
XFree86-tools-4.1.0-15.asp	Various tools for XFree86	linux/i386	
XFree86-twm-4.1.0-15.asp	A simple window manager	linux/i386	
XFree86-xdm-4.1.0-15.asp	X Display Manager	linux/i386	
XFree86-xf86cfg-4.1.0-15.asp	XFree86 configuration program	linux/i386	
XFree86-4.1.0-0.9.1	The basic fonts, programs and docs for an X workstation.	linux/k6	
XFree86-tools-4.1.0-0.9.1	Various tools for XFree86	linux/k6	
XFree86-twm-4.1.0-0.9.1	A simple window manager	linux/k6	
XFree86-xdm-4.1.0-0.9.1	X Display Manager	linux/k6	
XFree86-xf86cfg-4.1.0-0.9.1	XFree86 configuration program	linux/k6	
XFree86-4.0.2-6e	The basic fonts, programs and docs for an X workstation.	linux/ppc	
XFree86-tools-4.0.2-6e	Various tools for XFree86	linux/ppc	
XFree86-twm-4.0.2-6e	A simple window manager	linux/ppc	
XFree86-xdm-4.0.2-6e	X Display Manager	linux/ppc	
XFree86-xf86cfg-4.0.2-6e	XFree86 configurator	linux/ppc	
XFree86-Servers-3.3.6-44	XFree86 3.3.6 servers		
XFree86-Servers-3.3.6-43	XFree86 3.3.6 servers		
XFree86-Servers-3.3.6-38	XFree86 3.3.6 servers		
XFree86-3.3.6-29	The basic fonts, programs and docs for an X workstation.	linux/i386	linux/alpha linux/sparc
XFree86-3.3.6-20	The basic fonts, programs and docs for an X workstation.	Linux/sparc	Linux/i386 Linux/alpha
XFree86-3.3.6-11.2	The basic fonts, programs and docs for an X workstation.	Linux/ppc	
XFree86-3.3.5-1.6.0	The basic fonts, programs and docs for an X workstation.	Linux/i386	Linux/alpha Linux/sparc
XFree86-3.3.5-1.5.x	The basic fonts, programs and docs for an X workstation.	Linux/sparc	Linux/alpha Linux/i386
XFree86-3.3.5-1.4.x	The basic fonts, programs and docs for an X workstation.	Linux/sparc	Linux/alpha Linux/i386
XFree86-3.3.3.1-49a	Part of the XFree86 implementation of the X Window System.	Linux/ppc	Linux/ppc
	Part of the XFree86 implementation		

XFree86-3.3.3.1-37a	of the X Window System.	Linux/ppc	Linux/ppc	
XFree86-3.3.3.1-36nw10	Part of the XFree86 implementation of the X Window System.	Linux/armv4l		
XFree86-jpfonts-2.1-24	Japanese fixed fonts for X11.	linux/noarch	linux/noarch	
XFree86-100dpi-fonts-4.3.0-2.90.55	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.3.0-2.90.43	A set of 100dpi resolution fonts for the X Window System.	linux/i386	linux/i386	
XFree86-100dpi-fonts-4.3.0-2_nw3	A set of 100dpi resolution fonts for the X Window System.	linux/armv4l		
XFree86-100dpi-fonts-4.3.0-2	A set of 100dpi resolution fonts for the X Window System.	linux/i386	linux/i386	
XFree86-100dpi-fonts-4.2.1-23	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.2.1-13.73.23	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.2.0-72	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.2.0-8.6asp	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.2.0-8	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.2.0-6.30e	A set of 100dpi resolution fonts for the X Window System.	linux/ppc		
XFree86-100dpi-fonts-4.2.0-6.30d	A set of 100dpi resolution fonts for the X Window System.	linux/ppc		
XFree86-100dpi-fonts-4.1.0-50	A set of 100dpi resolution fonts for the X Window System.	linux/ia64	linux/i386	linux/i386
XFree86-100dpi-fonts-4.1.0-25	A set of 100dpi resolution fonts for the X Window System.	linux/ia64	linux/alpha	
XFree86-100dpi-fonts-4.1.0-15.asp	A set of 100dpi resolution fonts for the X Window System.	linux/i386		
XFree86-100dpi-fonts-4.0.2-6e	X Window System 100dpi fonts.	linux/ppc		
XFree86-100dpi-fonts-3.3.6-29	X Window System 100dpi fonts.	linux/sparc	linux/i386	linux/alpha
XFree86-100dpi-fonts-3.3.6-20	X Window System 100dpi fonts.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-100dpi-fonts-3.3.6-11.2	X Window System 100dpi fonts.	Linux/ppc		
XFree86-100dpi-fonts-3.3.5-1.6.0	X Window System 100dpi fonts.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-100dpi-fonts-3.3.5-1.5.x	X Window System 100dpi fonts.	Linux/sparc	Linux/alpha	Linux/i386

XFree86-100dpi-fonts-3.3.5-1.4.x	X Window System 100dpi fonts.	Linux/sparc	Linux/alpha	Linux/i386
XFree86-100dpi-fonts-3.3.3.1-49a	X Window System 100dpi fonts.	Linux/ppc	Linux/ppc	
XFree86-100dpi-fonts-3.3.3.1-37a	X Window System 100dpi fonts.	Linux/ppc	Linux/ppc	
XFree86-100dpi-fonts-3.3.3.1-36nw10	X Window System 100dpi fonts.	Linux/armv4l		
XFree86-75dpi-fonts-4.3.0-2.90.55	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.3.0-2.90.43	A set of 75dpi resolution fonts for the X Window System.	linux/i386	linux/i386	
XFree86-75dpi-fonts-4.3.0-2_nw3	A set of 75dpi resolution fonts for the X Window System.	linux/armv4l		
XFree86-75dpi-fonts-4.3.0-2	A set of 75dpi resolution fonts for the X Window System.	linux/i386	linux/i386	
XFree86-75dpi-fonts-4.2.1-23	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.2.1-13.73.23	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.2.0-72	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.2.0-8.6asp	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.2.0-8	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.2.0-6.30e	A set of 75dpi resolution fonts for the X Window System.	linux/ppc		
XFree86-75dpi-fonts-4.2.0-6.30d	A set of 75dpi resolution fonts for the X Window System.	linux/ppc		
XFree86-75dpi-fonts-4.1.0-50	A set of 75dpi resolution fonts for the X Window System.	linux/ia64	linux/i386	linux/i386
XFree86-75dpi-fonts-4.1.0-25	A set of 75dpi resolution fonts for the X Window System.	linux/alpha	linux/ia64	
XFree86-75dpi-fonts-4.1.0-15.asp	A set of 75dpi resolution fonts for the X Window System.	linux/i386		
XFree86-75dpi-fonts-4.1.0-0.9.1	A set of 75dpi resolution fonts for the X Window System.	linux/k6		
XFree86-75dpi-fonts-4.0.2-6e	A set of 75 dpi resolution fonts for the X Window System.	linux/ppc		
XFree86-75dpi-fonts-3.3.6-29	A set of 75 dpi resolution fonts for the X Window System.	linux/sparc	linux/i386	linux/alpha
XFree86-75dpi-fonts-3.3.6-20	A set of 75 dpi resolution fonts for the X Window System.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-75dpi-fonts-3.3.6-11.2	A set of 75 dpi resolution fonts for the X Window System.	Linux/ppc		

XFree86-75dpi-fonts-3.3.5-1.6.0	A set of 75 dpi resolution fonts for the X Window System.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-75dpi-fonts-3.3.5-1.5.x	A set of 75 dpi resolution fonts for the X Window System.	Linux/sparc	Linux/alpha	Linux/i386
XFree86-75dpi-fonts-3.3.5-1.4.x	A set of 75 dpi resolution fonts for the X Window System.	Linux/sparc	Linux/alpha	Linux/i386
XFree86-75dpi-fonts-3.3.3.1-49a	A set of 75 dpi resolution fonts for the X Window System.	Linux/ppc	Linux/ppc	
XFree86-75dpi-fonts-3.3.3.1-37a	A set of 75 dpi resolution fonts for the X Window System.	Linux/ppc	Linux/ppc	
XFree86-75dpi-fonts-3.3.3.1-36nw10	A set of 75 dpi resolution fonts for the X Window System.		Linux/armv4l	
XFree86-base-fonts-4.3.0-2.90.55	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.3.0-2.90.43	The collection of XFree86 core base fonts		linux/i386	linux/i386
XFree86-base-fonts-4.3.0-2_nw3	The collection of XFree86 core base fonts		linux/armv4l	
XFree86-base-fonts-4.3.0-2	The collection of XFree86 core base fonts		linux/i386	linux/i386
XFree86-base-fonts-4.2.1-23	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.2.1-13.73.23	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.2.0-72	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.2.0-8.6asp	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.2.0-8	The collection of XFree86 core base fonts		linux/i386	
XFree86-base-fonts-4.2.0-6.30e	The collection of XFree86 core base fonts		linux/ppc	
XFree86-base-fonts-4.2.0-6.30d	The collection of XFree86 core base fonts		linux/ppc	
XFree86-cyrillic-fonts-4.3.0-2.90.55	Cyrillic fonts for X.		linux/i386	
XFree86-cyrillic-fonts-4.3.0-2.90.43	Cyrillic fonts for X.		linux/i386	linux/i386
XFree86-cyrillic-fonts-4.3.0-2_nw3	Cyrillic fonts for X.		linux/armv4l	
XFree86-cyrillic-fonts-4.3.0-2	Cyrillic fonts for X.		linux/i386	linux/i386
XFree86-cyrillic-fonts-4.2.1-23	Cyrillic fonts for X.		linux/i386	
XFree86-cyrillic-fonts-4.2.1-13.73.23	Cyrillic fonts for X.		linux/i386	

XFree86-cyrillic-fonts-4.2.0-72	Cyrillic fonts for X.	linux/i386		
XFree86-cyrillic-fonts-4.2.0-8.6asp	Cyrillic fonts for X.	linux/i386		
XFree86-cyrillic-fonts-4.2.0-8	Cyrillic fonts for X.	linux/i386		
XFree86-cyrillic-fonts-4.2.0-6.30e	Cyrillic fonts for X.	linux/ppc		
XFree86-cyrillic-fonts-4.2.0-6.30d	Cyrillic fonts for X.	linux/ppc		
XFree86-cyrillic-fonts-4.1.0-50	Cyrillic fonts for X.	linux/ia64	linux/i386	linux/i386
XFree86-cyrillic-fonts-4.1.0-25	Cyrillic fonts for X.	linux/alpha	linux/ia64	
XFree86-cyrillic-fonts-4.1.0-15.asp	Cyrillic fonts for X.	linux/i386		
XFree86-cyrillic-fonts-4.1.0-0.9.1	Cyrillic fonts for X.	linux/k6		
XFree86-cyrillic-fonts-4.0.2-6e	Cyrillic fonts for X.	linux/ppc		
XFree86-cyrillic-fonts-3.3.6-29	Cyrillic fonts for X.	linux/sparc	linux/i386	linux/alpha
XFree86-cyrillic-fonts-3.3.6-20	Cyrillic fonts for X.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-cyrillic-fonts-3.3.6-11.2	Cyrillic fonts for X.	Linux/ppc		
XFree86-cyrillic-fonts-3.3.5-1.6.0	Cyrillic fonts for X.	Linux/sparc	Linux/i386	Linux/alpha
XFree86-cyrillic-fonts-3.3.5-1.5.x	Cyrillic fonts for X.	Linux/sparc	Linux/alpha	Linux/i386
XFree86-cyrillic-fonts-3.3.5-1.4.x	Cyrillic fonts for X.	Linux/sparc	Linux/alpha	Linux/i386
XFree86-cyrillic-fonts-3.3.3.1-49a	Cyrillic fonts - only needed on the server side.	Linux/ppc	Linux/ppc	
XFree86-cyrillic-fonts-3.3.3.1-37a	Cyrillic fonts - only needed on the server side.	Linux/ppc	Linux/ppc	
XFree86-cyrillic-fonts-3.3.3.1-36nw10	Cyrillic fonts - only needed on the server side.	Linux/armv4l		
XFree86-font-utils-4.3.0-2.90.55	Font utilities required for installing fonts	linux/i386		
XFree86-font-utils-4.3.0-2.90.43	Font utilities required for installing fonts	linux/i386	linux/i386	
XFree86-font-utils-4.3.0-2_nw3	Font utilities required for installing fonts	linux/armv4l		
XFree86-	Font utilities			

font-utils-4.3.0-2	required for installing fonts	linux/i386	linux/i386						
XFree86-font-utils-4.2.1-23	Font utilities required for installing fonts	linux/i386							
XFree86-font-utils-4.2.1-13.73.23	Font utilities required for installing fonts	linux/i386							
XFree86-font-utils-4.2.0-72	Font utilities required for installing fonts	linux/i386							
XFree86-font-utils-4.2.0-8.6asp	Font utilities required for installing fonts	linux/i386							
XFree86-font-utils-4.2.0-8	Font utilities required for installing fonts	linux/i386							
XFree86-font-utils-4.2.0-6.30e	Font utilities required for installing fonts	linux/ppc							
XFree86-font-utils-4.2.0-6.30d	Font utilities required for installing fonts	linux/ppc							
XFree86-ISO8859-9-2.1.2-14	Turkish language fonts and modmaps for X.	linux/noarch							
XFree86-ISO8859-9-2.1.2-10	Turkish language fonts and modmaps for X.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	
XFree86-ISO8859-9-2.1.2-9	Turkish language fonts and modmaps for X.	Linux/noarch	Linux/noarch						
XFree86-ISO8859-2-1.0-14	Central European language fonts for the X Window System.	linux/noarch							
XFree86-ISO8859-7-1.0-10	Greek language fonts for the X Window System.	linux/noarch	linux/noarch						
XFree86-ISO8859-2-1.0-9	Central European language fonts for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	
XFree86-ISO8859-2-1.0-8	Central European language fonts for the X	Linux/noarch	Linux/noarch						

	Window System.					
XFree86-ISO8859-7-1.0-8	Greek language fonts for the X Window System.	linux/noarch				
XFree86-ISO8859-7-1.0-4	Greek language fonts for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
XFree86-ISO8859-14-100dpi-fonts-4.3.0-2.90.55	ISO8859-14-100dpi-fonts	linux/i386				
XFree86-ISO8859-14-100dpi-fonts-4.3.0-2.90.43	ISO8859-14-100dpi-fonts	linux/i386	linux/i386			
XFree86-ISO8859-14-100dpi-fonts-4.3.0-2_nw3	ISO8859-14-100dpi-fonts	linux/armv4l				
XFree86-ISO8859-14-100dpi-fonts-4.3.0-2	ISO8859-14-100dpi-fonts		linux/i386	linux/i386		
XFree86-ISO8859-14-75dpi-fonts-4.3.0-2.90.55	ISO8859-14-75dpi-fonts		linux/i386			
XFree86-ISO8859-14-75dpi-fonts-4.3.0-2.90.43	ISO8859-14-75dpi-fonts		linux/i386	linux/i386		
XFree86-ISO8859-14-75dpi-fonts-4.3.0-2_nw3	ISO8859-14-75dpi-fonts		linux/armv4l			
XFree86-ISO8859-14-75dpi-fonts-4.3.0-2	ISO8859-14-75dpi-fonts		linux/i386	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.3.0-2.90.55	ISO8859-15-100dpi-fonts		linux/i386			
XFree86-ISO8859-15-100dpi-fonts-4.3.0-2.90.43	ISO8859-15-100dpi-fonts		linux/i386	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.3.0-2_nw3	ISO8859-15-100dpi-fonts		linux/armv4l			
XFree86-ISO8859-15-100dpi-fonts-4.3.0-2	ISO8859-15-100dpi-fonts		linux/i386	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.2.1-23	ISO8859-15-100dpi-fonts		linux/i386			
XFree86-ISO8859-15-100dpi-fonts-4.2.1-	ISO8859-15-100dpi-fonts		linux/i386			

13.73.23

XFree86-ISO8859-15-100dpi-fonts-4.2.0-72	ISO8859-15-100dpi-fonts	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.2.0-8.6asp	ISO8859-15-100dpi-fonts	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.2.0-8	ISO8859-15-100dpi-fonts	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.2.0-6.30e	ISO8859-15-100dpi-fonts	linux/ppc		
XFree86-ISO8859-15-100dpi-fonts-4.2.0-6.30d	ISO8859-15-100dpi-fonts	linux/ppc		
XFree86-ISO8859-15-100dpi-fonts-4.1.0-50	ISO8859-15-100dpi-fonts	linux/ia64	linux/i386	linux/i386
XFree86-ISO8859-15-100dpi-fonts-4.1.0-25	ISO8859-15-100dpi-fonts	linux/ia64	linux/alpha	
XFree86-ISO8859-15-100dpi-fonts-4.1.0-15.asp	ISO8859-15-100dpi-fonts	linux/i386		
XFree86-ISO8859-15-100dpi-fonts-4.1.0-0.9.1	ISO8859-15-100dpi-fonts	linux/k6		
XFree86-ISO8859-15-75dpi-fonts-4.3.0-2.90.55	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-75dpi-fonts-4.3.0-2.90.43	ISO8859-15-75dpi-fonts	linux/i386	linux/i386	
XFree86-ISO8859-15-75dpi-fonts-4.3.0-2_nw3	ISO8859-15-75dpi-fonts	linux/armv4l		
XFree86-ISO8859-15-75dpi-fonts-4.3.0-2	ISO8859-15-75dpi-fonts	linux/i386	linux/i386	
XFree86-ISO8859-15-75dpi-fonts-4.2.1-23	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-75dpi-fonts-4.2.1-13.73.23	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-75dpi-fonts-4.2.0-72	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-75dpi-fonts-4.2.0-8.6asp	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-	ISO8859-15-75dpi-fonts	linux/i386		

75dpi-fonts-4.2.0-8

XFree86-ISO8859-15-75dpi-fonts-4.2.0-6.30e	ISO8859-15-75dpi-fonts	linux/ppc		
XFree86-ISO8859-15-75dpi-fonts-4.2.0-6.30d	ISO8859-15-75dpi-fonts	linux/ppc		
XFree86-ISO8859-15-75dpi-fonts-4.1.0-50	ISO8859-15-75dpi-fonts	linux/ia64	linux/i386	linux/i386
XFree86-ISO8859-15-75dpi-fonts-4.1.0-25	ISO8859-15-75dpi-fonts	linux/alpha	linux/ia64	
XFree86-ISO8859-15-75dpi-fonts-4.1.0-15.asp	ISO8859-15-75dpi-fonts	linux/i386		
XFree86-ISO8859-15-75dpi-fonts-4.1.0-0.9.1	ISO8859-15-75dpi-fonts	linux/k6		
XFree86-ISO8859-2-100dpi-fonts-4.3.0-2.90.55	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-4.3.0-2.90.43	A set of 100dpi Central European language fonts for X.	linux/i386	linux/i386	
XFree86-ISO8859-2-100dpi-fonts-4.3.0-2_nw3	A set of 100dpi Central European language fonts for X.	linux/armv4l		
XFree86-ISO8859-2-100dpi-fonts-4.3.0-2	A set of 100dpi Central European language fonts for X.	linux/i386	linux/i386	
XFree86-ISO8859-2-100dpi-fonts-4.2.1-23	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-4.2.1-13.73.23	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-4.2.0-72	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-4.2.0-8.6asp	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-4.2.0-8	A set of 100dpi Central European language fonts for X.	linux/i386		
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/ppc		

4.2.0-6.30e	X.				
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/ppc			
4.2.0-6.30d	X.				
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/ia64	linux/i386	linux/i386	
4.1.0-50	X.				
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/alpha	linux/ia64		
4.1.0-25	X.				
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/i386			
4.1.0-15.asp	X.				
XFree86-ISO8859-2-100dpi-fonts-	A set of 100dpi Central European language fonts for	linux/k6			
4.1.0-0.9.1	X.				
XFree86-ISO8859-2-100dpi-fonts-	ISO 8859-2 fonts in 100 dpi resolution for the X Window System.	linux/ppc			
4.0.2-6e					
XFree86-ISO8859-2-100dpi-fonts-	ISO 8859-2 fonts in 100 dpi resolution for the X Window System.	linux/noarch			
1.0-14					
XFree86-ISO8859-2-100dpi-fonts-	ISO 8859-2 fonts in 100 dpi resolution for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
1.0-9					
XFree86-ISO8859-2-100dpi-fonts-	ISO 8859-2 fonts in 100 dpi resolution for the X Window System.	Linux/noarch	Linux/noarch		
1.0-8					
XFree86-ISO8859-2-75dpi-fonts-	A set of 75dpi Central European language fonts for	linux/i386			
4.3.0-2.90.55	X.				
XFree86-ISO8859-2-75dpi-fonts-	A set of 75dpi Central European language fonts for	linux/i386	linux/i386		

4.3.0-2.90.43 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/armv4l](#)
4.3.0-2_nw3 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#) [linux/i386](#)
4.3.0-2 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#)
4.2.1-23 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#)
4.2.1-
13.73.23 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#)
4.2.0-72 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#)
4.2.0-8.6asp X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/i386](#)
4.2.0-8 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/ppc](#)
4.2.0-6.30e X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/ppc](#)
4.2.0-6.30d X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/ia64](#) [linux/i386](#) [linux/i386](#)
4.1.0-50 X.

XFree86- A set of 75dpi
ISO8859-2- Central European
75dpi-fonts- language fonts for [linux/alpha](#) [linux/ia64](#)
4.1.0-25 X.

XFree86- A set of 75dpi
ISO8859-2- Central European

75dpi-fonts- 4.1.0-15.asp	language fonts for X.	linux/i386
XFree86- ISO8859-2- 75dpi-fonts- 4.1.0-0.9.1	A set of 75dpi Central European language fonts for X.	linux/k6
XFree86- ISO8859-2- 75dpi-fonts- 4.0.2-6e	A set of 75 dpi Central European language fonts for X.	linux/ppc
XFree86- ISO8859-2- 75dpi-fonts- 1.0-14	A set of 75 dpi Central European language fonts for X.	linux/noarch
XFree86- ISO8859-2- 75dpi-fonts- 1.0-9	A set of 75 dpi Central European language fonts for X.	Linux/noarch Linux/noarch Linux/noarch Linux/noarch Linux/noarch
XFree86- ISO8859-2- 75dpi-fonts- 1.0-8	A set of 75 dpi Central European language fonts for X.	Linux/noarch Linux/noarch
XFree86- ISO8859-2- Type1-fonts- 1.0-14	Type 1 scalable Central European language (ISO8859-2) fonts for X.	linux/noarch
XFree86- ISO8859-2- Type1-fonts- 1.0-9	Type 1 scalable Central European language (ISO8859-2) fonts for X.	Linux/noarch Linux/noarch Linux/noarch Linux/noarch Linux/noarch
XFree86- ISO8859-2- Type1-fonts- 1.0-8	Type 1 scalable Central European language (ISO8859-2) fonts for X.	Linux/noarch Linux/noarch
XFree86- ISO8859-7- 100dpi-fonts- 1.0-10	ISO 8859-7 fonts in 100 dpi resolution for the X Window System.	linux/noarch linux/noarch
XFree86- ISO8859-7- 100dpi-fonts- 1.0-8	ISO 8859-7 fonts in 100 dpi resolution for the X Window System.	linux/noarch
XFree86-	ISO 8859-7 fonts	

ISO8859-7- 100dpi-fonts- 1.0-4	in 100 dpi resolution for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
XFree86- ISO8859-7- 75dpi-fonts- 1.0-10	ISO 8859-7 fonts in 75 dpi resolution for the X Window System.	linux/noarch	linux/noarch			
XFree86- ISO8859-7- 75dpi-fonts- 1.0-8	ISO 8859-7 fonts in 75 dpi resolution for the X Window System.	linux/noarch				
XFree86- ISO8859-7- 75dpi-fonts- 1.0-4	ISO 8859-7 fonts in 75 dpi resolution for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
XFree86- ISO8859-7- Type1-fonts- 1.0-10	Type 1 scalable Greek (ISO 8859- 7) fonts	linux/noarch	linux/noarch			
XFree86- ISO8859-7- Type1-fonts- 1.0-8	Type 1 scalable Greek (ISO 8859- 7) fonts	linux/noarch				
XFree86- ISO8859-7- Type1-fonts- 1.0-4	Type 1 scalable Greek (ISO 8859- 7) fonts	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
XFree86- ISO8859-9- 100dpi-fonts- 4.3.0-2.90.55	ISO8859-9- 100dpi-fonts	linux/i386				
XFree86- ISO8859-9- 100dpi-fonts- 4.3.0-2.90.43	ISO8859-9- 100dpi-fonts	linux/i386	linux/i386			
XFree86- ISO8859-9- 100dpi-fonts- 4.3.0-2_nw3	ISO8859-9- 100dpi-fonts	linux/armv4l				
XFree86- ISO8859-9- 100dpi-fonts- 4.3.0-2	ISO8859-9- 100dpi-fonts	linux/i386	linux/i386			
XFree86-						

ISO8859-9- 100dpi-fonts- 4.2.1-23	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.1- 13.73.23	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.0-72	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.0-8.6asp	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.0-8	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.0-6.30e	ISO8859-9- 100dpi-fonts	linux/ppc		
XFree86- ISO8859-9- 100dpi-fonts- 4.2.0-6.30d	ISO8859-9- 100dpi-fonts	linux/ppc		
XFree86- ISO8859-9- 100dpi-fonts- 4.1.0-50	ISO8859-9- 100dpi-fonts	linux/ia64	linux/i386	linux/i386
XFree86- ISO8859-9- 100dpi-fonts- 4.1.0-25	ISO8859-9- 100dpi-fonts	linux/alpha	linux/ia64	
XFree86- ISO8859-9- 100dpi-fonts- 4.1.0-15.asp	ISO8859-9- 100dpi-fonts	linux/i386		
XFree86- ISO8859-9- 100dpi-fonts- 4.1.0-0.9.1	ISO8859-9- 100dpi-fonts	linux/k6		
XFree86- ISO8859-9- 100dpi-fonts- 2.1.2-14	100 dpi Turkish (ISO8859-9) fonts for X.	linux/noarch		

XFree86-ISO8859-9-100dpi-fonts-2.1.2-10	100 dpi Turkish (ISO8859-9) fonts for X.	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch	Linux/noarch
XFree86-ISO8859-9-100dpi-fonts-2.1.2-9	100 dpi Turkish (ISO8859-9) fonts for X.	Linux/noarch	Linux/noarch			
XFree86-ISO8859-9-75dpi-fonts-4.3.0-2.90.55	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-4.3.0-2.90.43	ISO8859-9-75dpi-fonts	linux/i386	linux/i386			
XFree86-ISO8859-9-75dpi-fonts-4.3.0-2_nw3	ISO8859-9-75dpi-fonts	linux/armv4l				
XFree86-ISO8859-9-75dpi-fonts-4.3.0-2	ISO8859-9-75dpi-fonts	linux/i386	linux/i386			
XFree86-ISO8859-9-75dpi-fonts-4.2.1-23	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-4.2.1-13.73.23	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-4.2.0-72	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-4.2.0-8.6asp	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-4.2.0-8	ISO8859-9-75dpi-fonts	linux/i386				
XFree86-ISO8859-9-75dpi-fonts-	ISO8859-9-75dpi-fonts	linux/ppc				

4.2.0-6.30e

XFree86-

ISO8859-9- ISO8859-9-

75dpi-fonts- 75dpi-fonts

[linux/ppc](#)

4.2.0-6.30d

XFree86-

ISO8859-9- ISO8859-9-

75dpi-fonts- 75dpi-fonts

[linux/ia64](#)

[linux/i386](#)

[linux/i386](#)

4.1.0-50

XFree86-

ISO8859-9- ISO8859-9-

75dpi-fonts- 75dpi-fonts

[linux/alpha](#)

[linux/ia64](#)

4.1.0-25

XFree86-

ISO8859-9- ISO8859-9-

75dpi-fonts- 75dpi-fonts

[linux/i386](#)

4.1.0-15.asp

XFree86-

ISO8859-9- ISO8859-9-

75dpi-fonts- 75dpi-fonts

[linux/k6](#)

4.1.0-0.9.1

XFree86-

ISO8859-9- 75 dpi Turkish

75dpi-fonts- (ISO8859-9) fonts [linux/noarch](#)

2.1.2-14 for X.

XFree86-

ISO8859-9- 75 dpi Turkish

75dpi-fonts- (ISO8859-9) fonts [Linux/noarch](#) [Linux/noarch](#) [Linux/noarch](#) [Linux/noarch](#) [Linux/noarch](#)

2.1.2-10 for X.

XFree86-ISO8859-9-

75dpi-fonts-2.1.2-9

75 dpi Turkish (ISO8859-9) fonts for X.

[Linux/noarch](#)

[Linux/noarch](#)

XFree86-KOI8-R-1.0-6

Russian and Ukrainian language fonts for the X Window System.

[linux/noarch](#)

[linux/noarch](#)

XFree86-KOI8-R-1.0-3

Russian and Ukrainian language fonts for the X Window System.

[linux/noarch](#)

XFree86-KOI8-R-100dpi-fonts-1.0-6

KOI8-R fonts in 100 dpi resolution for the X Window System.

[linux/noarch](#)

[linux/noarch](#)

XFree86-KOI8-R-100dpi-fonts-1.0-3

KOI8-R fonts in 100 dpi resolution for the X Window System.

[linux/noarch](#)

XFree86-KOI8-R-75dpi-fonts-1.0-6

A set of 75 dpi Russian and Ukrainian language fonts for X.

[linux/noarch](#)

[linux/noarch](#)

XFree86-KOI8-R-75dpi-fonts-1.0-3

A set of 75 dpi Russian and Ukrainian language fonts for X.

[linux/noarch](#)

XFree86-syriac-fonts-4.3.0-2.90.55

XFree86 Syriac TrueType fonts by Beth Mardutho

[linux/i386](#)

XFree86-syriac-fonts-4.3.0-2.90.43

XFree86 Syriac TrueType fonts by Beth Mardutho

[linux/i386](#)

[linux/i386](#)

XFree86-syriac-fonts-4.3.0-2_nw3	XFree86 Syriac TrueType fonts by Beth Mardutho	linux/armv4l	
XFree86-syriac-fonts-4.3.0-2	XFree86 Syriac TrueType fonts by Beth Mardutho	linux/i386	linux/i386
XFree86-truetype-fonts-4.3.0-2.90.55	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.3.0-2.90.43	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	linux/i386
XFree86-truetype-fonts-4.3.0-2_nw3	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/armv4l	
XFree86-truetype-fonts-4.3.0-2	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	linux/i386
XFree86-truetype-fonts-4.2.1-23	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.2.1-13.73.23	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.2.0-72	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.2.0-8.6asp	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.2.0-8	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/i386	
XFree86-truetype-fonts-4.2.0-6.30e	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/ppc	
XFree86-truetype-fonts-4.2.0-6.30d	XFree86 Luxi TrueType fonts by Bigelow and Holmes	linux/ppc	
XFree86-xtrap-clients-4.2.0-72	Xtrap client utilities	linux/i386	
XFree86-xtrap-clients-4.2.0-8.6asp	Xtrap client utilities	linux/i386	
XFree86-xtrap-clients-4.2.0-8	Xtrap client utilities	linux/i386	
XFree86-xtrap-clients-4.2.0-6.30e	Xtrap client utilities	linux/ppc	
XFree86-xtrap-clients-4.2.0-6.30d	Xtrap client utilities	linux/ppc	
xfree86_xkb_xml-0.4-1	XML-based XKB configuration registry	linux/noarch	
xfree86_xkb_xml-0.3-1	XML-based XKB configuration registry	linux/noarch	
xfree86_xkb_xml-0.2-1	XML-based XKB configuration registry	linux/noarch	
xinitrc-3.35-1	The default startup script for the X Window System.		
xinitrc-3.33-1	The default startup script for the X Window System.	linux/noarch	

xinitrc-3.32-1	The default startup script for the X Window System.	linux/noarch	linux/noarch	linux/noarch
xinitrc-3.31-1	The default startup script for the X Window System.	linux/noarch		
xinitrc-3.22-1	The default startup script for the X Window System.	linux/noarch		
xinitrc-3.20-1c	The default startup script for the X Window System.	linux/noarch		
xinitrc-3.20-1b	The default startup script for the X Window System.			
xinitrc-3.20-1	The default startup script for the X Window System.	linux/noarch		
xinitrc-2.9.10-1c	The default startup script for the X Window System.	linux/noarch		
xinitrc-2.9-1	The default startup script for the X Window System.	Linux/noarch	Linux/noarch	Linux/noarch
xinitrc-2.6-4	The default startup script for the X Window System.	Linux/noarch		
xinitrc-2.4.3-1	The default startup script for the X Window System.	Linux/noarch		
xinitrc-2.4.1-1	The default startup script for the X Window System.	Linux/noarch		
xscreensaver-3.24-1	X screen saver and locker	Linux/i386		
xscreensaver-gl-3.24-1	A set of GL screensavers	Linux/i386		
xsession-1.1-6	A program which controls your X Window System session.	Linux/sparc	Linux/i386	Linux/alpha
xterm-179-5	xterm terminal emulator for the X Window System			
xtermset-0.5.1-1	change some settings of an xterm via the command line	Linux/i386		
xtermset-0.5-1	change some settings of an xterm via the command line	Linux/i386		
xtt-fonts-0.19990222-9	Free Japanese TrueType fonts (mincho & gothic)	linux/noarch		
xtt-fonts-0.19990222-8.6x	Free Japanese TrueType fonts (mincho & gothic)	linux/noarch		
xvattr-1.3-fr2	Utility for getting and setting Xv attributes.	linux/i386		
xvattr-1.3-fr1	Utility for getting and setting Xv attributes.	linux/ppc	linux/i386	linux/ppc
xvim-0.2-1	X-Vietnamese Input Method	Linux/i586		

Generated by [rpm2html 1.8.1](#)

[Daniel Veillard](#), Thu Apr 15 16:05:03 2004

Index	index by Group	index by Distribution	index by Vendor	index by creation date	index by Name	Mirrors	Search
-----------------------	--------------------------------	---------------------------------------	---------------------------------	--	-------------------------------	-------------------------	------------------------

index by Vendor

- ["Chih-Wei Huang" <cwhuang@phys.ntu.edu.tw> \(4\)](#)
- ["Chih-Wei Huang" <cwhuang@trinova.com.tw> \(1\)](#)
- ["Markus Pilzecker" <mp@rhein-neckar.netsurf.de> \(37\)](#)
- ["Rui Miguel Seabra <rms@greymalkin.yi.org>" \(46\)](#)
- ["Soenke J.Peters" <peters@simprovement.com> \(2\)](#)
- ["William Evans" <william.evans@computer.org> \(4\)](#)
- [%packager \(4\)](#)
- [%{PACKAGER} \(6\)](#)
- [\(insert name here\) \(1\)](#)
- [<ebourg@po-box.mcgill.ca> \(2\)](#)
- [<source@mvista.com> \(2\)](#)
- [AJ Lewis <lewis@sistina.com> \(12\)](#)
- [ALzza Team <alzza@linux.sarang.net> \(39\)](#)
- [ASPLinux Packages Maintenance Team <packages@asp-linux.com> \(1\)](#)
- [ASPLinux Team <packages@asp-linux.com> \(4870\)](#)
- [Aaron Worley <aaron-w@earthling.net> \(7\)](#)
- [Abed M. Hammoud <abed.hammoud@medtronic.com> \(2\)](#)
- [Abel Cheung <deaddog@deaddog.org> \(734\)](#)
- [Abel Cheung <maddog@linux.org.hk> \(414\)](#)
- [Achim Steinmetz <as@ibm.net> \(1\)](#)
- [Adam \(StatiC\) Gibson <static@tampabay.rr.com> \(6\)](#)
- [Adam McKee \(3\)](#)
- [Adrian Ho <lexfiend@usa.net> \(7\)](#)
- [Adrian Likins <alikins@redhat.com> \(30\)](#)
- [Agustin Navarro <anp@cantv.net> \(42\)](#)
- [Akos Polster <akos@pipacs.com> \(5\)](#)
- [Akos Polster <akosp@freemail.hu> \(10\)](#)
- [Al Dev alavoor\[at\]yahoo.com \(2\)](#)
- [Alan Eldridge <alane@sourceforge.net> \(2\)](#)
- [Alan Eldridge <jalane@sourceforge.net> \(4\)](#)
- [Alan W. Irwin <irwin@beluga.phys.uvic.ca> \(18\)](#)
- [Alec Thomas <alecthomas@mail.com> \(2\)](#)
- [Aleksey Korenkov <akorenkov@users.sourceforge.net> \(2\)](#)
- [Aleksey Korenkov <alex@uznet.net> \(2\)](#)
- [Aleksey Sanin <aleksey@aleksey.com> \(131\)](#)
- [Alessandro Sabbatini <alsabbat@gulliver.unian.it> \(3\)](#)
- [Alex Agibalov \(2\)](#)
- [Alex B. Demko <ademko@mbnet.mb.ca> \(2\)](#)
- [Alex Izvorski <izv@together.net> \(2\)](#)
- [Alex Savguira <networks@ravdata.com> Magnus Stenman <stone@hkust.se>, Henri Gomez <gomez@slib.fr> \(3\)](#)
- [Alex Shnitman <alexsh@linux.org.il> \(1\)](#)
- [Alex Vorobiev <sasha@mathforum.com> \(5\)](#)

- [Alex deVries <adevries@engsoc.carleton.ca> \(10\)](#)
- [Alexander Feigl <Alexander.Feigl@gmx.de> \(8\)](#)
- [Alexander L. Belikoff \(abel@bfr.co.il\) \(4\)](#)
- [Alexander L. Belikoff \(abel@vallinor4.com\) \(2\)](#)
- [Alexander Skwar <ASkwar@DigitalProjects.com> \(374\)](#)
- [Alexandr Kanevskiy <kad@asplinux.ru> \(9\)](#)
- [Alexei Dets <dets@services.ru> \(2\)](#)
- [Alexey Nogin <ayn2@cornell.edu> \(82\)](#)
- [Alexey Nogin <nogin@dnrtm.ru> \(3\)](#)
- [Ali Onur Cinar, root@zdo.com \(2\)](#)
- [Allegro development team \(1\)](#)
- [Allegro development team <conductors@canvaslink.com> \(20\)](#)
- [Allen Rouse <stal@smoker.tx.symbio.net> \(2\)](#)
- [Amaury Amblard-Ladurantie <amaury@mandrakesoft.com> \(4\)](#)
- [Anders Karlsson <pugo@lysator.liu.se> \(1\)](#)
- [Anders Karlsson <pugo@pugo.org> \(2\)](#)
- [Anders Semb Hermansen <ahermans@vf.telia.no> \(2\)](#)
- [Andrea Borgia <borgia@cs.unibo.it> \(15\)](#)
- [Andrea Borgia <borgia@students.cs.unibo.it> \(22\)](#)
- [Andrea Pesarini <andreapes@iol.it> \(14\)](#)
- [Andrea Rizzi <rizzi@kde.org> \(4\)](#)
- [Andreas Bauer <baueran@users.sourceforge.net> \(22\)](#)
- [Andreas Krennmair <a.krennmair@aon.at> \(11\)](#)
- [Andrei Errapart <andreie@linux.ee> \(1\)](#)
- [Andrew E. Mileski <andrewm@netwinder.org> \(869\)](#)
- [Andrew Grimberg <tykeal-rpms@bardicgrove.org> \(5\)](#)
- [Andrew Grimberg <tykeal@bardicgrove.org> \(3\)](#)
- [Andrew Grimberg <tykeal@bigfoot.com> \(3\)](#)
- [Andrew J. Piziali <andy@piziali.dv.org> \(2\)](#)
- [Andrew Richards <ajr@users.sourceforge.net> \(12\)](#)
- [Andrew Skalski <askalski@buffalo.edu> \(2\)](#)
- [Andrew Snare \(earthpig@pigpond.com\) \(2\)](#)
- [Andrew Taylor <andrewtaylor@users.sourceforge.net> \(2\)](#)
- [Andrew Tridgell <tridge@samba.anu.edu.au> \(1\)](#)
- [Andrew Veliath <andrewtv@usa.net> \(9\)](#)
- [Andrew Veliath <veliaa@rpi.edu> \(5\)](#)
- [Andy C. Brandt <andy@cs.net.pl> \(1\)](#)
- [Andy C. Brandt <andy@uranos.eu.org> \(7\)](#)
- [Andy Dustman <andy@dustman.net> \(4\)](#)
- [Andy Lindeman <a_lindeman@hotmail.com> \(4\)](#)
- [Andy Loening <loening@ucla.edu> \(73\)](#)
- [Andy Piper <andy.piper@freeuk.com> \(3\)](#)
- [Anne-Marie Mahfouf \(2\)](#)
- [Anthony Rumble <anthony@infotainment.com.au> \(4\)](#)
- [Anthony Y.P. Wong <ypwong@alumni.cse.cuhk.edu.hk> \(2\)](#)
- [Anthony Y.P. Wong <ypwong@poboxes.com> \(16\)](#)
- [Antoine Ginies <aginies@mandrakesoft.com> \(198\)](#)
- [Anton Altaparmakov <aia21@cam.ac.uk> \(25\)](#)
- [Antoni Pamies Olive <toni@readysoft.net> \(45\)](#)
- [Antonio-Blasco Bonito <A.Bonito@cnuce.cnr.it> \(2\)](#)
- [Antti Andreimann <Antti.Andreimann@mail.ee> \(10\)](#)

- [Antti Andreimann <anttix@cyberix.edu.ee>](mailto:anttix@cyberix.edu.ee) (4)
- [Archit Baweja <bighead@crosswinds.net>](mailto:bighead@crosswinds.net) (6)
- [Ari Pollak <http://www.sourceware.net>](http://www.sourceware.net) (3)
- [Arie Tal <tal_arie@yahoo.com>](mailto:tal_arie@yahoo.com) (2)
- [Arien Malec <arien_malec@yahoo.com>](mailto:arien_malec@yahoo.com) (16)
- [Arix International <cpan2rpm@arix.com>](mailto:cpan2rpm@arix.com) (12)
- [Arjan Buursink <arjan.buursink@planet.nl>](mailto:arjan.buursink@planet.nl) (1)
- [Arkadi E. Shishlov <arkadi@kvin.lv>](mailto:arkadi@kvin.lv) (2)
- [Arkadi E. Shishlov <arkadi@kvin.lv>](mailto:arkadi@kvin.lv) (2)
- [Arkadiusz Miłkiewicz <misiek@debian.eu.org>](mailto:misiek@debian.eu.org) (14)
- [Arkadiusz Miłkiewicz <misiek@misiek.eu.org>](mailto:misiek@misiek.eu.org) (14)
- [Arnaud CALVO <arnaud@calvo-france.com>](mailto:arnaud@calvo-france.com) (4)
- [Arnaud Desmons <adesmons@mandrakesoft.com>](mailto:adesmons@mandrakesoft.com) (24)
- [Arnaud Gomes-do-Vale <arnaud@carrosse.frmug.org>](mailto:arnaud@carrosse.frmug.org) (15)
- [Arnaud Gomes-do-Vale <arnaud@glou.org>](mailto:arnaud@glou.org) (1)
- [Arnaud de Lorbeau <adelorbeau@mandrakesoft.com>](mailto:adelorbeau@mandrakesoft.com) (14)
- [Arne Coucheron <arneco@online.no>](mailto:arneco@online.no) (134)
- [Arthibus Gissehel <gissehel@vachefolle.com>](mailto:gissehel@vachefolle.com) (3)
- [Arturo Espinosa <arturo@nuclecu.unam.mx>](mailto:arturo@nuclecu.unam.mx) (1)
- [Arun Sharma <adsharma@home.com>](mailto:adsharma@home.com) (2)
- [Aurelien Bompard](#) (1)
- [Aurelien Lemaire <alemaire@mandrakesoft.com>](mailto:alemaire@mandrakesoft.com) (31)
- [Aurélien Gâteau <aurelien.gateau@mail.dotcom.fr>](mailto:aurelien.gateau@mail.dotcom.fr) (22)
- [Austin Acton <aacton@yorku.ca>](mailto:aacton@yorku.ca) (899)
- [Austin Acton <austin@linux.ca>](mailto:austin@linux.ca) (82)
- [Author -> Bjorn Lindgren \(packaged by Ryan Nash <posix@impulse.bitz.net>\)](#) (1)
- [Automatic](#) (23)
- [Avi Alkalay <avi@br.ibm.com>](mailto:avi@br.ibm.com) (4)
- [Axel von Arnim <vonarnim@club-internet.fr>](mailto:vonarnim@club-internet.fr) (4)
- [Ayub Yaqub, University College London, Centre for Nonlinear Dynamics & its Applications, London, WC1E 6BT <ayub.yaqub@ucl.ac.uk>](#) (2)
- [BSD](#) (1)
- [Baba Buehler <baba@uiuc.edu>](mailto:baba@uiuc.edu) (2)
- [Bart Whiteley <bart@caldera.com>](mailto:bart@caldera.com) (32)
- [Bellegarde Cedric <cedric.bellegarde@wanadoo.fr>](mailto:cedric.bellegarde@wanadoo.fr) (2)
- [Ben Escoto <bescoto@stanford.edu>](mailto:bescoto@stanford.edu) (2)
- [Ben Gordon <ben@bxg.org>](mailto:ben@bxg.org) (16)
- [Ben H Kram <jargon@modulo.org>](mailto:jargon@modulo.org) (4)
- [Ben Hochstedler <hochstrb@cs.rose-hulman.edu>](mailto:hochstrb@cs.rose-hulman.edu) (22)
- [Ben Hochstedler <hochstrb@cs.rose-hulman.edu>](mailto:hochstrb@cs.rose-hulman.edu) ; [Changjie Yang](#) (2)
- [Ben Martin](#) (4)
- [Ben Martin <monkeyiq@dingoblue.net.au>](mailto:monkeyiq@dingoblue.net.au) (15)
- [Ben Martin <monkeyiq@users.sourceforge.net>](mailto:monkeyiq@users.sourceforge.net) (37)
- [Ben Reed <ben@opennms.org>](mailto:ben@opennms.org) (4)
- [Ben Reser <ben@reser.org>](mailto:ben@reser.org) (148)
- [Ben Reser <breser@mandrakesecure.net>](mailto:breser@mandrakesecure.net) (6)
- [Ben Stanley <bds02@uow.edu.au>](mailto:bds02@uow.edu.au) (3)
- [Ben Woodard <ben@valinux.com>](mailto:ben@valinux.com) (15)
- [Benjamin Gordon <ben@bxg.org>](mailto:ben@bxg.org) (45)
- [Benjamin Meyer <ben@meyerhome.net>](mailto:ben@meyerhome.net) (5)
- [Bernhard Fastenrath <fasten@shw.com>](mailto:fasten@shw.com) (6)

- [Bernhard Herzog <bernhard@users.sourceforge.net>](#) (18)
- [Bernhard Herzog <sketch@online.de>](#) (1)
- [Bernhard J. Pietsch <bjtp@gmx.net>](#) (36)
- [Bernhard Walle <Bernhard.Walle@gmx.de>](#) (2)
- [Bert de Bruijn <bob@ccl.kuleuven.ac.be>](#) (6)
- [Bibek Sahu <scorpio@dodds.net>](#) (2)
- [Bill Gatliff <bgat@billgatliff.com>](#) (1)
- [Bill Kawakami <billy@cafe.net>](#) (4)
- [Bill Nottingham <wen1@cec.wustl.edu>](#) (8)
- [Bill Wilson <bill@gkrellm.net>](#) (2)
- [Bishop Clark \(LC957\) <bishop@platypus.bc.ca>](#) (24)
- [Björn Sahlström](#) (7)
- [Björn Sahlström <bjorn@kbear.org>](#) (2)
- [Bob Deblrier <bob@virtualunlimited.com>](#) (38)
- [Bob Smith <bob@thestuff.net>](#) (74)
- [Bob Tanner <tanner@real-time.com>](#) (39)
- [Borek Lupomesky <Borek.Lupomesky@ujep.cz>](#) (4)
- [Brad Dixon <gt5392c@prism.gatech.edu>](#) (2)
- [Bradley A. Smith <bradley@baysmith.com>](#) (8)
- [Brandon Craig Rhodes <bcrhodes@cc.gatech.edu>](#) (6)
- [Brandon Craig Rhodes <bcrhodes@cc.gatech.edu>](#) (4)
- [Brandon George](#) (5)
- [Brandon Lees <brandon2@users.sourceforge.net>](#) (16)
- [Breyten J. Ernsting <bje@dds.nl>](#) (4)
- [Brian Bruns <bruns@Magenet.com>](#) (37)
- [Brian Candler <B.Candler@pobox.com>](#) (7)
- [Brian Craft <bcboy@dorothy.wanglab.brandeis.edu>](#) (1)
- [Brian Duggan <bduggan@oven.com>](#) (8)
- [Brian Finley <brian@baldguysoftware.com>](#) (9)
- [Brian L. Johnson <blj8@blj8.com>](#) (1)
- [Brian Langenberger <brian@biosci.umn.edu>](#) (6)
- [Brian Masney <masneyb@seul.org>](#) (1)
- [Brian T.N. Gunney](#) (4)
- [Brian T.N. Gunney <btng@users.sourceforge.net>](#) (8)
- [Brian Wright](#) (4)
- [Brian Wright <bdw at briandwright dot com>](#) (9)
- [Brian Wright <bdw@aturna.com>](#) (22)
- [Brian Wright <bdw@aturna.con>](#) (2)
- [Brian Wright <bdw@aturna.net>](#) (3)
- [Brook Humphrey <bah@webmedic.net>](#) (10)
- [Brook Humphrey <bah@webmedic.net>](#), [Glen Lee Edwards <glen@fcwm.org>](#) (1)
- [Brook Humphrey <bah@webmedic.net>](#), [Joachim Ansorg <jansorg@gmx.de>](#) (23)
- [Bruce Guenter <bruce.guenter@qcc.sk.ca>](#) (21)
- [Bruce Guenter <bruceg@em.ca>](#) (23)
- [Bruce Kall <kall@compass.com>](#) (2)
- [Bruce McDonald <bruce@triph0p.dyn.ml.org http://triph0p.dyn.ml.org>](#) (1)
- [Bruce Warren Bigby <bbigby@alum.MIT.edu>](#) (2)
- [Bruce Winter](#) (4)
- [Bruno Coudoin <bcoudoin.coudoin@free.fr>](#) (64)
- [Bruno Coudoin <bruno.coudoin@free.fr>](#) (1)
- [Bruno Pires Marinho <bapm@netcabo.pt>](#) (12)

- [Bryce Denney <bryce.denney@bigfoot.com> \(18\)](#)
- [Buchan Milne <bgmilne@linux-mandrake.com> \(414\)](#)
- [Buchan Milne <buchan@linux-mandrake.com> \(2\)](#)
- [Builder <builder@greymalkin.yi.org> \(7\)](#)
- [Byron Young \(6\)](#)
- [Byron Young <spamiccling@yahoo.com> \(5\)](#)
- [C C Magnus Gustavsson <http://magnus.gustavsson.se/> \(6\)](#)
- [C. Scott Ananian <cananian@alumni.princeton.edu> \(25\)](#)
- [Calvin Arndt <calarndt@yahoo.com> \(3\)](#)
- [Camille Begnis <camille@mandrakesoft.com> \(99\)](#)
- [Carles Arjona <nospammer@bigfoot.com> \(16\)](#)
- [Carlo Marcelo Arenas Belon <carenas@chasqui.lared.net.pe> \(2\)](#)
- [Carlo Marcelo Arenas Belon <carenas@sajino.lared.net.pe> \(6\)](#)
- [Carsten Müller <malison@malison.de> \(2\)](#)
- [Carsten Pfeiffer <pfeiffer@kde.org> \(5\)](#)
- [Catalin Climov <xxl@thekompany.com> \(9\)](#)
- [CeeJay Computer Services \(1\)](#)
- [Cesar Cardoso <ccardoso@cesarcardoso.ddns.org> \(3\)](#)
- [Cesar Cardoso <ccardoso@cesarcardoso.dyn.ml.org> \(2\)](#)
- [Charles 'Buck' Krasic <krasic@acm.org> \(24\)](#)
- [Charles <int@linuxcore.com> \(18\)](#)
- [Charles A Edwards <eslrahc@bellsouth.net> \(53\)](#)
- [Charles A Edwards <eslrahc@mandrake.org> \(35\)](#)
- [Charles Duffy <duffy@bigfoot.com> \(2\)](#)
- [Charles Kerr <charles@rebelbase.com> \(2\)](#)
- [Charlie Zender <zender@uci.edu> \(8\)](#)
- [Chih-Wei Huang <cwhuang@cle.linux.org.tw> \(4\)](#)
- [Chih-Wei Huang <cwhuang@linux.org.tw> \(1\)](#)
- [Chih-Wei Huang <cwhuang@trinova.com.tw> \(1\)](#)
- [Chmouel Boudjnah <chmouel@mandrakesoft.com> \(63\)](#)
- [Chris <cfreeze@cs.clemson.edu> \(1\)](#)
- [Chris <cfreeze@mystic.cs.clemson.edu> \(2\)](#)
- [Chris Allegretta <chrisa@asty.org> \(19\)](#)
- [Chris Ausbrooks <weed@bucket.pp.ua.r.edu> \(105\)](#)
- [Chris Ausbrooks <weed@thebucket.org> \(16\)](#)
- [Chris Black <cblack@bunkai.com> \(2\)](#)
- [Chris Cogdon <chris@felidae.apana.org.au> \(6\)](#)
- [Chris Dent <cdent@kiva.net> \(6\)](#)
- [Chris Green \(2\)](#)
- [Chris Green <cmg@uab.edu> \(9\)](#)
- [Chris Green <sprout@dok.org> \(13\)](#)
- [Chris Grossmann <Chris.Grossmann@duke.edu> \(4\)](#)
- [Chris Grossmann <chris_grossmann@bigfoot.com> \(8\)](#)
- [Chris Hamilton <chrish@realminfo.com> \(4\)](#)
- [Chris Lindell <clindell@users.sourceforge.net> \(9\)](#)
- [Chris Schantz \(2\)](#)
- [Chris Schoeneman <crs23@bigfoot.com> \(2\)](#)
- [Chris Spell <spell@wilmington.net> \(3\)](#)
- [Christian Belisle <cbelisle@mandrakesoft.com> \(10\)](#)
- [Christian Biesinger <biesi@users.sourceforge.net> \(2\)](#)
- [Christian Gavin <cgavin@dnai.com> \(7\)](#)

- [Christian Hammond <chipx86@gnupdate.org> \(19\)](#)
- [Christian Hammond <chipx86@portaldesign.net> \(38\)](#)
- [Christian Joensson <chj@lin.foa.se> \(5\)](#)
- [Christian Joensson FOA 72 <chj@lin.foa.se> \(10\)](#)
- [Christian Kreibich <christian@whoop.org> \(6\)](#)
- [Christian Mauduit <ufoot@ufoot.org> \(4\)](#)
- [Christian Mayrhuber <Christian.Mayrhuber@gmx.net> \(2\)](#)
- [Christian Schaller <Uraeus@linuxrising.com> \(3\)](#)
- [Christian Schaller <Uraeus@linuxrising.org> \(74\)](#)
- [Christian Schaller <Uraues@linuxrising.org> \(6\)](#)
- [Christian Schulte <schulte@ps.uni-sb.de> \(4\)](#)
- [Christoph Lorenz <ChLorenz@csi.com> \(2\)](#)
- [Christoph Lorenz <Christoph_Lorenz@gmx.de> \(32\)](#)
- [Christophe BERNARD <bernard@cmm.ensmp.fr> \(13\)](#)
- [Christophe Bernard <bernard@cmapx.polytechnique.fr> \(12\)](#)
- [Christophe Prud'homme <prudhomm@users.sourceforge.net> \(15\)](#)
- [Christophe Tronche <ch.tronche@computer.org> \(5\)](#)
- [Christopher Blizzard <blizzard@appliedtheory.com> \(3\)](#)
- [Christopher Blizzard <blizzard@redhat.com> \(9\)](#)
- [Christopher Craig <ccraig@ccraig.org> \(2\)](#)
- [Christopher Fisk <chrisf@atecone.net> \(2\)](#)
- [Christopher McCrory <chrismcc@netus.com> \(4\)](#)
- [Christopher Yeoh <cyeh@linuxcare.com> \(2\)](#)
- [Chuck Mead <csm@moongroup.com> \(1\)](#)
- [Claudiu Costin <claudiuc@geocities.com> \(4\)](#)
- [Clic-dev <clic-dev@mandrakesoft.com> \(5\)](#)
- [Clinton Work <work@scripty.com> \(6\)](#)
- [Clive Stubbings \(4\)](#)
- [Colin Leroy <colin@colino.net> \(3\)](#)
- [Colin Phipps <cph@cph.demon.co.uk> \(2\)](#)
- [Colin Phipps <cphipp@doomworld.com> \(3\)](#)
- [Conrad Steenberg <conrad@hep.caltech.edu> \(4\)](#)
- [Conrad Steenberg <conrad@srl.caltech.edu> \(4\)](#)
- [Conrad Steenberg <conrad@srl.caltech.edu>, Alan W. Irwin <irwin@uvastro.phys.uvic.ca> \(3\)](#)
- [Cons Maintainer <cons-maintainer@gnu.org> \(3\)](#)
- [CoreDmp@users.sourceforge.net \(8\)](#)
- [Corey Minyard <minyard@acm.org> \(2\)](#)
- [Craig Franklin <craigfranklin@home.com> \(4\)](#)
- [Craig Kattner <ckattner@sun.iwu.edu> \(2\)](#)
- [Craig Lanning <clanning@users.sourceforge.net> \(4\)](#)
- [Cris Bailiff \(2\)](#)
- [Cristian Martinez <cfuga@itam.mx> \(11\)](#)
- [Cristian Paslaru <crys@dntis.ro> \(2\)](#)
- [Crossfire Development Team <crossfire-devel@lists.real-time.com> \(75\)](#)
- [Curtis Ireland <cireland@aurora.carleton.ca> \(2\)](#)
- [Cyclic Software \(4\)](#)
- [D. Barbier, Christophe Bernard <{barbier,bernard}@cmapx.polytechnique.fr> \(13\)](#)
- [DU Xiaoming <dxiaoming@mandrakesoft.com> \(6\)](#)
- [Dag Wieers <dag@wieers.com> \(83\)](#)
- [Daisuke SUZUKI <daisuke@linux.or.jp> \(2\)](#)
- [Dale Bewley <dale@bewley.net> \(6\)](#)

- [Dale K. Hawkins <dhawkins@cdrmts.com> \(3\)](#)
- [Damian Christey <daemonc@netscape.net> \(4\)](#)
- [Damian Ivereigh <damian@cisco.com> \(13\)](#)
- [Damien Chaumette <dchaumette@mandrakesoft.com> \(43\)](#)
- [Damien Krotkine <dams@tuxfamily.org> \(4\)](#)
- [Damien Miller <damien@ibs.com.au> \(2\)](#)
- [Damien Miller <djm@ibs.com.au> \(3\)](#)
- [Damien Miller <djm@mindrot.org> \(42\)](#)
- [Damien Miller <dmiller@ilogic.com.au> \(2\)](#)
- [Damien Miller <dmiller@vitnet.com.sg> \(64bit fix by: Hugo van der Kooij <hvdtkooij@caiwn.nl>\) \(1\)](#)
- [Dan Bethe <dan_bethe@yahoo.com> \(1\)](#)
- [Dan E. Anderson <http://dan.drydog.com/> \(10\)](#)
- [Dan E. Anderson <http://www.dan.pmbc.com/> \(4\)](#)
- [Dan E. Kelley <Dan.Kelley@Dal.CA> \(53\)](#)
- [Dan Fandrich <dan@cone harvesters.com> \(1\)](#)
- [Dan Fandrich <dan@fch.wimsey.bc.ca> \(3\)](#)
- [Dan Johnson <dj51d@progworks.net> \(2\)](#)
- [Dan Johnson <dj51d@warbirdsurvivors.org> \(6\)](#)
- [Dan Lark <dlark@elmresources.com> \(3\)](#)
- [Dan Mueth <d-mueth@uchicago.edu> \(4\)](#)
- [Daniel Caujolle-Bert <flrmb@users.sourceforge.net> \(18\)](#)
- [Daniel Deimert <d1dd@dtek.chalmers.se> \(1\)](#)
- [Daniel Hanks <hanksdc@plug.org> \(2\)](#)
- [Daniel Resare <noa@metamatrix.se> \(1\)](#)
- [Daniel Schepler <dks2@cec.wustl.edu> \(12\)](#)
- [Daniel Serodio <dserodio@email.com> \(4\)](#)
- [Daniel Serodio <dserodio@email.com>, Sourav K. Mandal <Sourav.Mandal@ikaran.com> \(5\)](#)
- [Daniel Spiljar <dspiljar@public.srce.hr> \(4\)](#)
- [Daniel Sully <daniel-rpms@electricrain.com> \(4\)](#)
- [Daniel Veillard <veillard@w3.org> \(121\)](#)
- [Daniel.Veillard@w3.org \(6\)](#)
- [Daniele Arduini <d.arduini@rn.nettuno.it> \(2\)](#)
- [Daniele Berti <http://space.tin.it/computer/wvtberti/index.htm> \(2\)](#)
- [Danilo Caio Marcucci Marques \(danpos at if dot uff dot br\) \(2\)](#)
- [Danny Sung <dannys@mail.com> \(19\)](#)
- [Danny Tholen <obiwan@mailmij.org> \(14\)](#)
- [Daouda Lo <daouda@mandrakesoft.com> \(151\)](#)
- [Darin Fisher <darinf@users.sourceforge.net> \(27\)](#)
- [Darin Fisher <dfisher@jagger.berkeley.edu> \(1\)](#)
- [Dario Abatianni <eisfuchs@tigress.com> \(8\)](#)
- [Darren B. Hiebert <darren@hiebert.com> \(6\)](#)
- [Darren B. Hiebert <http://darren.hiebert.com> \(10\)](#)
- [Darren Nickerson <darren@dazza.org> \(3\)](#)
- [Darren Nickerson <darren@info.tpc.int> \(1\)](#)
- [Dave Beckett <Dave.Beckett@bristol.ac.uk> \(96\)](#)
- [Dave Cook <dmcook@cts.com> \(3\)](#)
- [Dave Jones <davej@suse.de> \(14\)](#)
- [Dave Pearson <davep@davep.org> \(3\)](#)
- [Dave Peticolas <dave@krondo.com> \(52\)](#)
- [Dave Woodhouse <Dave@imladris.demon.co.uk> \(18\)](#)

- [David A. Parker <david@neongoat.com>](mailto:david@neongoat.com) (15)
- [David Axmark <david@mysql.com>](mailto:david@mysql.com) (341)
- [David BAUDENS <baudens@mandrakesoft.com>](mailto:baudens@mandrakesoft.com) (48)
- [David Beattie <dbeattie@softhome.net>](mailto:dbeattie@softhome.net) (1)
- [David Benson <daveb@ffem.org>](mailto:daveb@ffem.org) (3)
- [David Chappell <ppr-bugs@trincoll.edu>](mailto:ppr-bugs@trincoll.edu) (1)
- [David Clark <silenus@telus.net>](mailto:silenus@telus.net) (3)
- [David D. Kilzer <ddkilzer@earthlink.net>](mailto:ddkilzer@earthlink.net) (3)
- [David E. Myers <dem@skyline.rtp.nc.us.com>](mailto:dem@skyline.rtp.nc.us.com) (2)
- [David Essex <dessex@arvotek.net>](mailto:dessex@arvotek.net) (14)
- [David Essex <dessex@nospam.arvotek.net.nospam>](mailto:dessex@nospam.arvotek.net.nospam) (2)
- [David F. Elliott <http://www.infinite-internet.net/~dfe/>](http://www.infinite-internet.net/~dfe/) (14)
- [David F. Elliott <http://www.thecomputerbuilder.com/~dfe/>](http://www.thecomputerbuilder.com/~dfe/) (7)
- [David H. Munro <munro1@llnl.gov>](mailto:munro1@llnl.gov) (6)
- [David Haller <david@dhaller.de>](mailto:david@dhaller.de) (6)
- [David Hayes <david@hayes-family.org>](mailto:david@hayes-family.org) (11)
- [David Hedbor <david@hedbor.org>](mailto:david@hedbor.org) (8)
- [David Ishee <dmi1@ra.msstate.edu>](mailto:dmi1@ra.msstate.edu) (4)
- [David Jao <djao@dominia.mit.edu>](mailto:djao@dominia.mit.edu) (4)
- [David Jao <djao@dominia.org>](mailto:djao@dominia.org) (4)
- [David Kuestler <dkuestler@australia.or.jp>](mailto:dkuestler@australia.or.jp) (2)
- [David Kuestler <kuestler@zeta.org.au>](mailto:kuestler@zeta.org.au) (8)
- [David M. Cook, <dmcook@cts.com>](mailto:dmcook@cts.com) (2)
- [David MacKenzie <djm@web.us.uu.net>](mailto:djm@web.us.uu.net) (3)
- [David Moore <dcm@acm.org>](mailto:dcm@acm.org) (16)
- [David Nedrow <dnedrow@usa.net>](mailto:dnedrow@usa.net) (15)
- [David Slimp <dslimp@locg.com>](mailto:dslimp@locg.com) (1)
- [David Sugar <dyfet@ostel.com>](mailto:dyfet@ostel.com) (42)
- [David Sugar <dyfet@tycho.com>](mailto:dyfet@tycho.com) (30)
- [David Sweet <dsweet@chaos.umd.edu>](mailto:dsweet@chaos.umd.edu) (1)
- [David Walluck <david@anti-microsoft.org>](mailto:david@anti-microsoft.org) (319)
- [David Walluck <david@anti-microsoft.org>](mailto:david@anti-microsoft.org) (8)
- [David Walluck <walluck@linux-mandrake.com>](mailto:walluck@linux-mandrake.com) (9)
- [David Walluck <walluck@mandrake-linux.com>](mailto:walluck@mandrake-linux.com) (18)
- [Davide Angelocola <davide178@inwind.it>](mailto:davide178@inwind.it) (2)
- [Davide Libenzi <davidel@xmailserver.org>](mailto:davidel@xmailserver.org) (2)
- [Dax Games <dgames@isoc.net>](mailto:dgames@isoc.net) (2)
- [Dean Johnson <dtj@sgi.com>](mailto:dtj@sgi.com), [Michael Jennings <mej@valinux.com>](mailto:mej@valinux.com), [The Rasterman \(Carsten Haitzler\) <raster@valinux.com>](#) and [Zac Sprackett <zsprackett@valinux.com>](mailto:zsprackett@valinux.com) (50)
- [Dejan Lekic <leka@9mb.net>](mailto:leka@9mb.net) (17)
- [Dejan Lekic <leka@linuks.org>](mailto:leka@linuks.org) (1)
- [Del <del@babel.com.au>](mailto:del@babel.com.au) (25)
- [Dennis Furey <fureyd@sbu.ac.uk>](mailto:fureyd@sbu.ac.uk) (2)
- [Dennis Heltzel <tiki-rpm@fishroom.net>](mailto:tiki-rpm@fishroom.net) (2)
- [Derald Metzger <dmetz@remotesensing.org>](mailto:dmetz@remotesensing.org) (2)
- [Derrick Hamner <derrickh@ntiusa.com>](mailto:derrickh@ntiusa.com) (1)
- [Derry Bryson <derry@techass.com>](mailto:derry@techass.com) (23)
- [Devin <devin@vibechild.com>](mailto:devin@vibechild.com) (11)
- [Devrim SERAL <devrim@tef.gazi.edu.tr>](mailto:devrim@tef.gazi.edu.tr) (2)
- [Dick Porter <dick@cymru.net>](mailto:dick@cymru.net) (3)
- [Didi Baer <didi@servus.at>](mailto:didi@servus.at) (2)

- [Dimitri Papadopoulos <dpo@club-internet.fr> \(12\)](#)
- [DindinX <odin@mandrakesoft.com> \(2\)](#)
- [Dirk Koopman <djk@tobit.co.uk> \(5\)](#)
- [Dirk Mueller \(1\)](#)
- [Dmitry V. Levin <ldv@fandra.org> \(14\)](#)
- [Dom Lachowicz <dominicl@seas.upenn.edu> \(4\)](#)
- [Dominik Mierzejewski <dmierzej@elka.pw.edu.pl> \(8\)](#)
- [Dominik Mierzejewski <dominik@rangers.eu.org> \(197\)](#)
- [Donovan Rebbechi <elflord@pegasus.rutgers.edu> \(4\)](#)
- [Doug Sisk sisk@wix.com \(1\)](#)
- [Doug Sisk sisk@wix.com, Henri Gomez nri@mail.dotcom.fr \(11\)](#)
- [Douglas N. Arnold <dna@psu.edu> \(1\)](#)
- [Dr. Henrik Seidel <seidel@mpimg-berlin-dahlem.mpg.de> \(7\)](#)
- [Dragos Acostachioaie <dragos@iname.com> \(11\)](#)
- [Dries Verachtert <dries@ace.ulyssis.student.kuleuven.ac.be> \(2\)](#)
- [Dries Verachtert <dries@ulyssis.org> \(140\)](#)
- [Duke Nukem <duke@diktyocafe.gr> \(1\)](#)
- [Duncan <duncan_haldane@users.sourceforge.net> \(3\)](#)
- [Duncan Haldane <f.d.m.haldane@mciworld.com> \(4\)](#)
- [Easy Software Products \(1\)](#)
- [Ed Schaller <schallee@akopia.com> \(10\)](#)
- [Eddie Dunckley <eddie@obsidian.co.za> \(8\)](#)
- [Eddie Kohler <eddietwo@lcs.mit.edu> \(3\)](#)
- [Eduardo Israel Osorio Hernández <iosorio@punto-deventa.com> \(1\)](#)
- [Eduardo M Kalinowski \(ekalin@iname.com\) \(2\)](#)
- [Edward S. Marshall <emarshal@logic.net> \(17\)](#)
- [Edwin Chan \(7\)](#)
- [Edwin Huffstutler <edwinh@computer.org> \(7\)](#)
- [Edwin Huffstutler <flexbackup@home.com> \(2\)](#)
- [Elliot Lee <sopwith@cuc.edu> \(4\)](#)
- [Emanuele Cisbani <cisba@tiscali.it> \(2\)](#)
- [Emanuele Cisbani <cisba@tiscalinet.it> \(4\)](#)
- [Emile Heyns <emile@iris-advies.com> \(5\)](#)
- [Emmanuel Jeandel <E_Jeandel@mail.dotcom.fr> \(2\)](#)
- [Emmanuel ZURMELY <emmanuelz@users.sourceforge.net> \(20\)](#)
- [Endre Zelena <ezelena@mazsola.buvi.matav.hu> \(2\)](#)
- [Eric S. Raymond <esr@thyrsus.com> \(10\)](#)
- [Eric Schendel <esche@users.sourceforge.net> \(3\)](#)
- [Erick Gallezio <eg@unice.fr> \(4\)](#)
- [Erik Nolf \(25\)](#)
- [Erik.Kunze@fantasy.muc.de \(2\)](#)
- [Erwan Velu \(erwan@mandrakesoft.com\) \(86\)](#)
- [Erwan Velu <erwan@mandrakesoft.com> \(3\)](#)
- [Erwin Waterlander <waterlan@xs4all.nl> \(2\)](#)
- [Eskil Heyn Olsen <deity@eskil.dk> \(5\)](#)
- [Ethan Blanton <eblanton@cs.purdue.edu> \(25\)](#)
- [Etienne Faure <etienne@mandrakesoft.com> \(7\)](#)
- [Eugen C. <eug@thekompany.com> \(44\)](#)
- [Eugen Constantinescu <eug@thekompany.com> \(1\)](#)
- [Everett Holland <everett@roanoke.infi.net> \(2\)](#)
- [Eze Ogwuma <ghost@typhoon.dircon.co.uk> \(6\)](#)

- [Eze Ogwuma <typhoon@dircon.co.uk>](mailto:typhoon@dircon.co.uk) (4)
- [FURUSAWA, Kazuhisa <furusawa@com.cs.osakafu-u.ac.jp>](mailto:furusawa@com.cs.osakafu-u.ac.jp) (1)
- [FVWM Themes Developers <fvwm-themes-devel@lists.sourceforge.net>](mailto:fvwm-themes-devel@lists.sourceforge.net) (22)
- [FVWM Workers <fvwm-workers@fvwm.org>](mailto:fvwm-workers@fvwm.org) (73)
- [Fedor G. Pikus <pikus@iname.com>](mailto:pikus@iname.com) (2)
- [Felipe Bergo <bergo@seul.org>](mailto:bergo@seul.org) (18)
- [Felix Braun <fbraun@atdot.org>](mailto:fbraun@atdot.org) (1)
- [Fernando Lunardelli <howardc@creepyc0.8m.com>](mailto:howardc@creepyc0.8m.com) (1)
- [Fire Dragon <fdragon@nosferatu.world-net.net>](mailto:fdragon@nosferatu.world-net.net) (2)
- [Florian Wallner <wallner@speed-link.de>](mailto:wallner@speed-link.de) (1)
- [Florin <florin@mandrakesoft.com>](mailto:florin@mandrakesoft.com) (341)
- [Florin Andrei <florin@sgi.com>](mailto:florin@sgi.com) (10)
- [Florin Grad <fgrad@mandrakesoft.com>](mailto:fgrad@mandrakesoft.com) (35)
- [Francesco Bresciani <fbresciani@users.sourceforge.net>](mailto:fbresciani@users.sourceforge.net) (27)
- [Francis G. McCabe <fgm@fla.fujitsu.com>](mailto:fgm@fla.fujitsu.com) (15)
- [Francis Galiegue <fg@mandrakesoft.com>](mailto:fg@mandrakesoft.com) (1)
- [Francis Galiegue <francis@mandrakesoft.com>](mailto:francis@mandrakesoft.com) (1)
- [Francis J. Lacoste <flacoste@logreport.org>](mailto:flacoste@logreport.org) (1)
- [Francis J. Lacoste <francis@Contre.COM>](mailto:francis@Contre.COM) (1)
- [Francisco Vides Fernandez <pvides@dedalo-ing.com>](mailto:pvides@dedalo-ing.com) (4)
- [Fran ck Villaume <fvill@freesurf.fr>](mailto:fvill@freesurf.fr) (58)
- [Fran ois Dupoux <fdupoux@partimage.org>](mailto:fdupoux@partimage.org) (19)
- [Frank DENIS <j@pureftpd.org>](mailto:j@pureftpd.org) (118)
- [Frank Dauer <f@paf.net>](mailto:f@paf.net) (2)
- [Frank Schwanz <schwanz@fh-brandenburg.de>](mailto:schwanz@fh-brandenburg.de) (4)
- [Frank Sweetser <rasmusin@wpi.edu>](mailto:rasmusin@wpi.edu) (2)
- [Frank V. Castellucci <frankc@users.sourceforge.net>](mailto:frankc@users.sourceforge.net) (30)
- [Frank de Lange <frank@unternet.org>](mailto:frank@unternet.org) (7)
- [Franklin Chen <Franklin.Chen@cs.cmu.edu>](mailto:Franklin.Chen@cs.cmu.edu) (10)
- [Franklin Chen <chen+@cs.cmu.edu>](mailto:chen+@cs.cmu.edu) (2)
- [Fran ois Dupoux <fdupoux@free.fr>](mailto:fdupoux@free.fr) (10)
- [Fran ois Dupoux <fdupoux@voila.fr>](mailto:fdupoux@voila.fr) (2)
- [Fran ois PONS <fpons@mandrakesoft.com>](mailto:fpons@mandrakesoft.com) (405)
- [Frederic Lepied <Frederic.Lepied@sugix.fr>](mailto:Frederic.Lepied@sugix.fr) (1)
- [Frederic Lepied <flepied@mandrakesoft.com>](mailto:flepied@mandrakesoft.com) (131)
- [Frederik Bijlsma <fbijlsma@redhat.de>](mailto:fbijlsma@redhat.de) (5)
- [Frederik Bijlsma <freddi@atdot.org>](mailto:freddi@atdot.org) (2)
- [Frederik Bilhaut <bilhaut_f@mail.cpod.fr>](mailto:bilhaut_f@mail.cpod.fr) (19)
- [Frediano Ziglio <freddy77@angelfire.com>](mailto:freddy77@angelfire.com) (13)
- [Fritz Elfert <fritz.elfert@to.com>](mailto:fritz.elfert@to.com) (63)
- [Fryguy_ <fryguy@falehope.co>](mailto:fryguy@falehope.co) (3)
- [Fryguy_ <fryguy@falsehope.com>](mailto:fryguy@falsehope.com) (18)
- [Fuhito Suguri <bitwalk@jcom.home.ne.jp>](mailto:bitwalk@jcom.home.ne.jp) (10)
- [Fuhito Suguri <bitwalk@nyc.odn.ne.jp>](mailto:bitwalk@nyc.odn.ne.jp) (6)
- [G. Thomas <g.thomas@opengroup.org>](mailto:g.thomas@opengroup.org), [O. Fourdan <fourdan@xfce.org>](mailto:fourdan@xfce.org), [P. Reich <reichp@ameritech.net>](mailto:reichp@ameritech.net) (1)
- [GGZ Dev Team <ggz-dev@lists.sourceforge.net>](mailto:ggz-dev@lists.sourceforge.net) (6)
- [GNOME Packaging Project <http://bugzilla.gnome.org/>](http://bugzilla.gnome.org/) (132)
- [Gabriel Cain](#) (2)
- [Gabriel Gerhardsson <mr.g@home.se>](mailto:mr.g@home.se) (131)
- [Gabriele Giorgetti <g.gabriele@europe.com>](mailto:g.gabriele@europe.com) (1)

- [Ganesh Sittampalam <ganesh@earth.li> \(7\)](#)
- [Garen Erdoisa <scamper@trisk.com> \(7\)](#)
- [Gary Thomas <g.thomas@opengroup.org> \(1\)](#)
- [Gary Zharzhavsky <Gary.Zharzhavsky@us.matranet.com> \(6\)](#)
- [Gavin Hurlbut <gjhurlbu@users.sourceforge.net> \(12\)](#)
- [Geoffrey Lee <snailtalk@mandrakesoft.com> \(56\)](#)
- [George Bronnikov <goga@bertie.la.utexas.edu> \(8\)](#)
- [George Foot <george.foot@merton.oxford.ac.uk> \(14\)](#)
- [George Kraft IV <gk4@us.ibm.com> \(8\)](#)
- [Gerald Teschl <gerald@esi.ac.at> \(2\)](#)
- [Gerhard Häring <gerhard@bigfoot.de> \(4\)](#)
- [Germain Garand <germain@ebooksfrance.org> \(4\)](#)
- [Gernot Tenchio <gernot@tenchicolor.de> \(2\)](#)
- [Gilles Detillieux <grdetil@scrc.umanitoba.ca> \(4\)](#)
- [Gilles Foucault <gilles.foucault@wanadoo.fr> \(8\)](#)
- [Giuseppe Borzi' <gborzi@ieee.org> \(4\)](#)
- [Giuseppe Ghibò <ghibo@mandrakesoft.com> \(411\)](#)
- [Giuseppe Paterno' <gpaterno@spacelab.gpaterno.com> \(2\)](#)
- [Glen Wiley <gwiley@ieee.org> \(8\)](#)
- [Gomez Henri <gomez@slib.fr> \(15\)](#)
- [Gomez Henri <hgomez@slib.fr> \(8\)](#)
- [Gomez Henri <hgomez@slib.fr>, Henner Zeller <zeller@to.com> \(3\)](#)
- [Gomez Henri <hgomez@slib.fr>, Tim Verhoeven <dj@sin.khk.be> \(2\)](#)
- [Gomez Henri <hgomez@slib.fr>, Troels Arvin <troels@arvin.dk> \(2\)](#)
- [Gomez Henri <hgomez@users.sourceforge.net>, Tim Verhoeven <dj@sin.khk.be> \(4\)](#)
- [Goneri Le Bouder <glebouder@mandrakesoft.com> \(8\)](#)
- [Gordon Machel <gmachel@users.sourceforge.net> \(70\)](#)
- [Gordon Machel <gmachel@users.sourceforge.net>, el-cuco <cuco3001@yahoo.com> \(2\)](#)
- [Gordon Rowell <gordonr@e-smith.com> \(2\)](#)
- [Gordon Tyler <gordon@doxxx.net> \(2\)](#)
- [Goulven Le Jeune <goulven.le-jeune@bull.net> \(2\)](#)
- [Graeme S. Roy <graeme@epc.co.uk> \(2\)](#)
- [Graham Roff \(37\)](#)
- [Graham Todd <gtodd@yorku.ca> \(1\)](#)
- [Grant Beattie <grant@spyda.net> \(2\)](#)
- [Greg Boehnlein <damin@nacs.net> \(3\)](#)
- [Greg LaPolla <glapolla@earthlink.net> \(3\)](#)
- [Griff Miller <griff.miller@positron.com> \(6\)](#)
- [Grzegorz Mucha <grzes@twins.pk.edu.pl> \(3\)](#)
- [Guardian Digital, Inc. <http://www.guardiandigital.com> \(1799\)](#)
- [Guardian Digital, Inc. <http://www.linuxseclabs.com> \(53\)](#)
- [Guido Draheim <guidod@gmx.de> \(158\)](#)
- [Guido Draheim <guidod@linux-mandrake.com> \(12\)](#)
- [Guillaume Cottenceau <gc@mandrakesoft.com> \(397\)](#)
- [Guillaume Rousse <g.rousse@linux-mandrake.com> \(345\)](#)
- [Guillaume Rousse <guillomovitch@linux-mandrake.com> \(182\)](#)
- [Guillaume Rousse <guillomovitch@mandrake.org> \(75\)](#)
- [Guillaume Rousse <guillomovitch@users.sourceforge.net> \(1\)](#)
- [Guillermo Castro <gcastro@users.sourceforge.net> \(2\)](#)
- [Gustavo Niemeyer <gustavo@nn.com.br> \(8\)](#)
- [Gustavo Niemeyer <niemeyer@conectiva.com> \(6\)](#)

- [Gwenole Beauchesne <gbeauchesne@mandrakesoft.com>](mailto:gbeauchesne@mandrakesoft.com) (1440)
- [Gyepi Sam <gyepi@peanutpress.com>](mailto:gyepi@peanutpress.com) (1)
- [Göran Uddeborg <goeran@uddeborg.pp.se>](mailto:goeran@uddeborg.pp.se) (8)
- [Göran Uddeborg <göran@uddeborg.pp.se>](mailto:göran@uddeborg.pp.se) (3)
- [Göran Weinholt <weinholt@linux.nu>](mailto:weinholt@linux.nu) (3)
- [Götz Waschk <waschk@linux-mandrake.com>](mailto:waschk@linux-mandrake.com) (2216)
- [Götz Waschk <waschk@mandrakesoft.com>](mailto:waschk@mandrakesoft.com) (227)
- [H. Peter Anvin <hpa@zytor.com>](mailto:hpa@zytor.com) (1)
- [H. Wade Minter <minter@lunenburg.org>](mailto:minter@lunenburg.org) (7)
- [HA Quoc-Viet <viet@mandrakesoft.com>](mailto:viet@mandrakesoft.com) (64)
- [Haavard Kvaalen <havardk@netcom.no>](mailto:havardk@netcom.no) (7)
- [Haavard Kvaalen <havardk@xmms.org>](mailto:havardk@xmms.org) (7)
- [Habibie <habibie@NetZero.net>](mailto:habibie@NetZero.net) (8)
- [Hakan Tandogan <hakan@iconsult.com>](mailto:hakan@iconsult.com) (10)
- [Han Boetes <han@linux-mandrake.com>](mailto:han@linux-mandrake.com) (144)
- [Han Boetes <han@mijncomputer.nl>](mailto:han@mijncomputer.nl) (1)
- [Hans Grobler <grobh@sun.ac.za>](mailto:grobh@sun.ac.za) (1)
- [Hans Zoebelein <hzo@goldfish.cube.net>](mailto:hzo@goldfish.cube.net) (26)
- [Hao Li <hli@wag.caltech.edu>](mailto:hli@wag.caltech.edu) (27)
- [Harald Welte <laforge@gnumonks.org>](mailto:laforge@gnumonks.org) (9)
- [Hari Nair <hari@alumni.caltech.edu>](mailto:hari@alumni.caltech.edu) (31)
- [Harry Henry Gebel <hgebel@inet.net>](mailto:hgebel@inet.net) (2)
- [Havoc Pennington <hp@redhat.com>](mailto:hp@redhat.com) (4)
- [Hawk, John <visionary@gtemail.net>](mailto:visionary@gtemail.net) (10)
- [Heitham Omar <super-ice@ntlworld.com>](mailto:super-ice@ntlworld.com) (4)
- [Heitham Omar <super_ice@ntlworld.com>](mailto:super_ice@ntlworld.com) (3)
- [Helio Chissini de Castro <helio@conectiva.com.br>](mailto:helio@conectiva.com.br) (358)
- [Helmut Michels <michels@linmpi.mpg.de>](mailto:michels@linmpi.mpg.de) (16)
- [Heng Yuan <heng@ag.arizona.edu>](mailto:heng@ag.arizona.edu) (2)
- [Henner Zeller <zeller@to.com>](mailto:zeller@to.com) (2)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr) (35)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr) and [William Stearns <wstearns@pobox.com>](mailto:wstearns@pobox.com) (2)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [Aron Griffis <agriffis@css.tayloru.edu>](mailto:agriffis@css.tayloru.edu) (2)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [Aron Griffis <agriffis@css.tayloru.edu>](mailto:agriffis@css.tayloru.edu), [Avi Alkalay <avi@br.ibm.com>](mailto:avi@br.ibm.com) (10)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [Aron Griffis <agriffis@css.tayloru.edu>](mailto:agriffis@css.tayloru.edu), [Avi Alkalay <avi@br.ibm.com>](mailto:avi@br.ibm.com), [Andrew Pendray <andrew.s.pendray@us.pwcglobal.com>](mailto:andrew.s.pendray@us.pwcglobal.com) (5)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [Murray Todd Williams <MurrayTodd@yahoo.com>](mailto:MurrayTodd@yahoo.com) (10)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [Ragnar Kjørstad <rrdtool@ragnark.vestdata.no>](mailto:rrdtool@ragnark.vestdata.no) (3)
- [Henri Gomez <gomez@slib.fr>](mailto:gomez@slib.fr), [William Stearns <wstearns@pobox.com>](mailto:wstearns@pobox.com), and [Dave Wreski <dave@linuxsecurity.com>](mailto:dave@linuxsecurity.com), [Wim Vandersmissen <wim@bofh.be>](mailto:wim@bofh.be) (6)
- [Henri Gomez <hgomez@slib.fr>](mailto:hgomez@slib.fr) (25)
- [Henri Gomez <hgomez@users.sourceforge.net>](mailto:hgomez@users.sourceforge.net) (43)
- [Henri Gomez <hgomez@users.sourceforge.net>](mailto:hgomez@users.sourceforge.net), [Ragnar Kjørstad <rrdtool@ragnark.vestdata.no>](mailto:rrdtool@ragnark.vestdata.no) (3)
- [Henri Gomez <hgomez@users.sourceforge.net>](mailto:hgomez@users.sourceforge.net), [Riku Meskanen <mesrik@iki.fi>](mailto:mesrik@iki.fi) (19)
- [Henrik Brix Andersen <btech1@users.sourceforge.net>](mailto:btech1@users.sourceforge.net) (2)
- [Henrik Johnson <tora@underscore.se>](mailto:tora@underscore.se) (31)
- [Henrik Seidel <seidel@mpimg-berlin-dahlem.mpg.de>](mailto:seidel@mpimg-berlin-dahlem.mpg.de) (3)
- [Henrik Steen <dat94hsn@idt.mdh.se>](mailto:dat94hsn@idt.mdh.se) (3)
- [Herbert Valerio Riedel <hvr@gnu.org>](mailto:hvr@gnu.org) (8)

- [Herwin Jan Steehouwer <steehouwer@kde.org>](mailto:steehouwer@kde.org) (12)
- [Hio <hio@lombardiacom.it>](mailto:hio@lombardiacom.it) (16)
- [Hiroyuki Yamamoto <hiro-y@kcn.ne.jp>](mailto:hiro-y@kcn.ne.jp) (2)
- [HoFi <hofi@WiZARD.DunaNet.hu>](mailto:hofi@WiZARD.DunaNet.hu) (6)
- [Holger DÜrer <holly@starship.python.net>](mailto:holly@starship.python.net) (2)
- [Holger Klemm](#) (1)
- [Hollis Blanchard <hollis@penguinppc.org>](mailto:hollis@penguinppc.org) (11)
- [Hollis Blanchard <hollis@yellowdoglinux.com>](mailto:hollis@yellowdoglinux.com) (4)
- [Honest Pete's Packaging <peter@accsys.demon.co.uk>](mailto:peter@accsys.demon.co.uk) (1)
- [Horms <horms@valinux.com>](mailto:horms@valinux.com) (2)
- [Horms <horms@verge.net.au>](mailto:horms@verge.net.au) (19)
- [Horms <horms@vergenet.net>](mailto:horms@vergenet.net) (30)
- [Horms <horms@zipworld.com.au>](mailto:horms@zipworld.com.au) (2)
- [Hugo Monteiro <hmmm@fct.unl.pt>](mailto:hmmm@fct.unl.pt) (2)
- [Hugo van der Kooij](#) (104)
- [Hugo van der Kooij <hvdtkooij@caiw.nl>](mailto:hvdtkooij@caiw.nl) (15)
- [ICS <info@ics.com>](mailto:info@ics.com) (8)
- [ITZA Sdn. Bhd. <www.itzaberhad.com>](http://www.itzaberhad.com), [Alexey Zilber <itzaberhad@yahoo.com>](mailto:itzaberhad@yahoo.com) (1)
- [Iain Holmes](#) (1)
- [Ian A Cameron <i.a.cameron@open.ac.uk>](mailto:i.a.cameron@open.ac.uk) (2)
- [Ian K <iguy@ionsphere.org>](mailto:iguy@ionsphere.org) (9)
- [Ian Koenig <iguy@ionsphere.org>](mailto:iguy@ionsphere.org) (4)
- [Ian Macdonald <ian@caliban.org>](mailto:ian@caliban.org) (83)
- [Ian Macdonald <ianmacd@caliban.org>](mailto:ianmacd@caliban.org) (4)
- [Ian Macdonald <ianmacd@xs4all.nl>](mailto:ianmacd@xs4all.nl) (69)
- Irene.Vatton@w3.org (4)
- [Ivan Fernandez <ivan.fernandez@vanderbilt.edu>](mailto:ivan.fernandez@vanderbilt.edu) (2)
- [Ivo Panacek <ivop@jilabs.cz>](mailto:ivop@jilabs.cz) (2)
- [Iñaki García Etxebarria <garetxe@users.sourceforge.net>](mailto:garetxe@users.sourceforge.net) (41)
- [J. Ali Harlow <ali@avrc.city.ac.uk>](mailto:ali@avrc.city.ac.uk) (20)
- [J. Ali Harlow <ali@juiblex.co.uk>](mailto:ali@juiblex.co.uk) (12)
- [J.E. Hoffmann <je-h@gmx.net>](mailto:je-h@gmx.net) (1)
- [Jack Lloyd <lloyd@acm.jhu.edu>](mailto:lloyd@acm.jhu.edu) (29)
- [Jaco Greeff <jaco@mandrake.org>](mailto:jaco@mandrake.org) (7)
- [Jade](#) (26)
- [Jaime Silva <mono@andromeda.utp.edu.co>](mailto:mono@andromeda.utp.edu.co) (2)
- [Jakob 'sparky' Kaivo <jkaivo@nodomainname.net>](mailto:jkaivo@nodomainname.net) (6)
- [Jakob Kaivo <jake@nodomainname.net>](mailto:jake@nodomainname.net) (5)
- [Jakub Jelinek <jj@ultra.linux.cz>](mailto:jj@ultra.linux.cz) (2)
- [James A. Pattie <james@pcxperience.com>](mailto:james@pcxperience.com) (126)
- [James D. Lucha <lucha@pe.net>](mailto:lucha@pe.net) (5)
- [James Dean Palmer <james@tiger-marmalade.com>](mailto:james@tiger-marmalade.com) (6)
- [James Graves <ansible@xnet.com>](mailto:ansible@xnet.com) (4)
- [James Henstridge](#) (12)
- [James Henstridge <james@daa.com.au>](mailto:james@daa.com.au) (72)
- [James Henstridge <james@daa.com.au>](mailto:james@daa.com.au) (Rel 1), [John Wingate <johnww@worldpath.net>](mailto:johnww@worldpath.net) (Rel 2) (2)
- [James Juran <juran@cse.psu.edu>](mailto:juran@cse.psu.edu) (2)
- [James Su <suzhe@gnuchina.org>](mailto:suzhe@gnuchina.org) (2)
- [James Youngman <jay@gnu.org>](mailto:jay@gnu.org) (2)
- [Jamie Cameron](#) (12)

- [Jan "Yenya" Kasprzak <kas@fi.muni.cz> \(57\)](#)
- [Jan Borsodi <jb@ez.no> \(16\)](#)
- [Jan Gentsch <gentsch@ifm.uni-hamburg.de> \(176\)](#)
- [Jan Kratochvil <short@ucw.cz> \(10\)](#)
- [Jan Oberländer <mindriot@gmx.net> \(4\)](#)
- [Jan Prikryl <prikryl@acm.org> \(3\)](#)
- [Jan Prokop <jprokop@ibl.sk> \(3\)](#)
- [Jan Rochat <rochat@aeteurope.nl> \(6\)](#)
- [Jan Schaumann <jschauma@netmeister.org> \(6\)](#)
- [Jan Vicherek <rpm@ied.com> \(5\)](#)
- [Jani Hakala <jahakala@cc.jyu.fi> \(5\)](#)
- [Jaroslaw Niewegłowski <janie@mat.uni.torun.pl> \(2\)](#)
- [Jason Gilbert <jason@scott.net> \(7\)](#)
- [Jason Kasper \(vanRijn\) <vR@movingparts.net> \(6\)](#)
- [Jason Spangler <jasons@usemail.com> \(46\)](#)
- [Jason Spence <thalakan@technologist.com> \(2\)](#)
- [Jason T. Collins <dark@ksu.edu> \(2\)](#)
- [Jasper Berlijn <jasper@stack.nl> \(2\)](#)
- [Jasta <jasta@gotlinux.org> \(8\)](#)
- [Java Software <jre-comments@java.sun.com> \(1\)](#)
- [Jay Beale <jay@mandrakesoft.com> \(95\)](#)
- [Jay Sekora <js@aq.org> \(2\)](#)
- [Jean Christophe ANDRÉ <Jean-Christophe.ANDRE@aupelf.refer.org> \(2\)](#)
- [Jean Delvare <delvare@ensicaen.ismra.fr> \(10\)](#)
- [Jean-Eric Cuendet <cuendet@linkvest.ch> \(2\)](#)
- [Jean-Luc Fontaine <jfontain@free.fr> \(5\)](#)
- [Jean-Marc Valin \(valj01@gel.usherb.ca\) \(22\)](#)
- [Jean-Michel Dault <jmdault@mandrakesoft.com> \(377\)](#)
- [Jean-Paul Smets <jp@nexedi.com> \(4\)](#)
- [Jean-Sebastien Morisset <jsmoriss@mvlan.net> \(6\)](#)
- [Jedi j@4u.net \(73\)](#)
- [Jeff Breidenbach <jeff@jab.org> \(2\)](#)
- [Jeff Dike <jdike@karaya.com> \(16\)](#)
- [Jeff Garzik <jgarzik@mandrakesoft.com> \(16\)](#)
- [Jeff Gehlbach <jeffg@pobox.com> \(3\)](#)
- [Jeff Rush <jrush@summit-research.com> \(1\)](#)
- [Jeff Rush <jrush@taupro.com> \(74\)](#)
- [Jeff Stern <jstern@eclectic.ss.uci.edu> \(3\)](#)
- [Jeff Uphoff <juphoff@nrao.edu> \(1\)](#)
- [Jeffrey Bakker <staer@email.com> \(1\)](#)
- [Jeffrey Watts <watts@sunflower.com> \(2\)](#)
- [Jens Leuschner <leuschner@gmx.net> \(18\)](#)
- [Jens Mueller <jm30@inf.tu-dresden.de> \(2\)](#)
- [Jens Müller <jm30@inf.tu-dresden.de> \(2\)](#)
- [Jere Knaappila <jeremk@users.sourceforge.net> \(2\)](#)
- [Jeremie Petit \(petit@eurecom.fr\) \(1\)](#)
- [Jeremy Doolin <jdoolin@cs.ohio.edu> \(2\)](#)
- [Jeremy Doolin <jdoolin@cs.ohiou.edu> \(9\)](#)
- [Jeremy Fischer <jeremy@pobox.com>, Gomez Henri <gomez@slib.fr>, Rod Cordova <rcordova@ethernet.org> \(2\)](#)
- [Jeremy Hansen <jeremy@xxedgexx.com> \(1\)](#)

- [Jeremy Hylton <jeremy@beopen.com>](mailto:jeremy@beopen.com) (2)
- [Jeremy Wise <jwise@pathwaynet.com>](mailto:jwise@pathwaynet.com) (32)
- [Jerome Couderc \(j.couderc@ifrance.com\)](mailto:j.couderc@ifrance.com) (5)
- [Jerome Couderc <j.couderc@ifrance.com>](mailto:j.couderc@ifrance.com) (94)
- [Jess Mahan <jess@codehost.com>](mailto:jess@codehost.com) (2)
- [Jesse B. Off <joff@iastate.edu>](mailto:joff@iastate.edu) (2)
- [Jesse Keating <hosting@j2solutions.net>](mailto:hosting@j2solutions.net) (1)
- [Jesse Kuang <kjx@mandrakesoft.com>](mailto:kjx@mandrakesoft.com) (1)
- [Jesus Ortega \(a.k.a. Nitebirdz\) <nitebirdz@openboxgroup.com>](mailto:nitebirdz@openboxgroup.com) (2)
- [Jhon H. Caicedo <jhcaiced@osso.org.co>](mailto:jhcaiced@osso.org.co) (89)
- [Jim Capp <jcapp@anteil.com>](mailto:jcapp@anteil.com) (5)
- [Jim Capp <jcapp@anteil.com>](mailto:jcapp@anteil.com), [Jay Mumper <jdm@anteil.com>](mailto:jdm@anteil.com) (3)
- [Jim Garrison <garrison@users.sourceforge.net>](mailto:garrison@users.sourceforge.net) (2)
- [Joachim Ansorg <jansorg@gmx.de>](mailto:jansorg@gmx.de) (4)
- [Jochem Wichers Hoeth <wiho@chem.uva.nl>](mailto:wiho@chem.uva.nl) (19)
- [Jochen Schmitt <Jochen@herr-schmitt.de>](mailto:Jochen@herr-schmitt.de) (2)
- [Joe Bowman <jbowman@kiva.net>](mailto:jbowman@kiva.net) (8)
- [Joe Laffey <software@laffeycomputer.com>](mailto:software@laffeycomputer.com) (2)
- [Joel D. Elkins <jde@binarts.com>](mailto:jde@binarts.com) (2)
- [Joel Sloan <jjs@pobox.com>](mailto:jjs@pobox.com) (6)
- [Joel Young <jyoung@erols.com>](mailto:jyoung@erols.com) (11)
- [Joerg Dorchain <joerg@dorchain.net>](mailto:joerg@dorchain.net) (6)
- [Joerg Fritsch <joerg.fritsch@planet-interkom.de>](mailto:joerg.fritsch@planet-interkom.de) [<joerg.fritsch@tesion.de>](mailto:joerg.fritsch@tesion.de) (6)
- [Joerg Fritsch <joerg.fritsch@planet-interkom.de>](mailto:joerg.fritsch@planet-interkom.de) [<joerg.fritsch@tesion.de>](mailto:joerg.fritsch@tesion.de) (4)
- [Johan Bevemyr <jb@erix.ericsson.se>](mailto:jb@erix.ericsson.se) (23)
- [Johan Rossouw <johanrossouw@users.sourceforge.net>](mailto:johanrossouw@users.sourceforge.net) (5)
- [Johan Rossouw <jrossouw@alcatel.altech.co.za>](mailto:jrossouw@alcatel.altech.co.za) (6)
- [Johan Vromans <jvromans@squirrel.nl>](mailto:jvromans@squirrel.nl) (4)
- [Johan Walles \(d92-jwa@nada.kth.se\)](mailto:d92-jwa@nada.kth.se) (3)
- [Johannes Tevessen <j.tevessen@gmx.net>](mailto:j.tevessen@gmx.net) (2)
- [John A. Martin <jam@jamux.com>](mailto:jam@jamux.com) (7)
- [John Allen](#) (6)
- [John C. Peterson <jaypee@netcom.com>](mailto:jaypee@netcom.com) (4)
- [John Dennis <jdennis@redhat.com>](mailto:jdennis@redhat.com) (7)
- [John Donoghue <donoghue@chariot.net.au>](mailto:donoghue@chariot.net.au) (13)
- [John Donoghue <jade@dell.bard>](mailto:jade@dell.bard) (5)
- [John Ellis <gqview@email.com>](mailto:gqview@email.com) (1)
- [John Ellis <gqview@geocities.com>](mailto:gqview@geocities.com) (1)
- [John Farrell <Friendless@users.sourceforge.net>](mailto:Friendless@users.sourceforge.net) (4)
- [John H Terpstra \[Samba-Team\] <jht@samba.anu.edu.au>](mailto:jht@samba.anu.edu.au) (1)
- [John Harper <john@dcs.warwick.ac.uk>](mailto:john@dcs.warwick.ac.uk) (189)
- [John Knottenbelt <jak@fla.fujitsu.com>](mailto:jak@fla.fujitsu.com) (4)
- [John Morrissey <jwm@proftpd.org>](mailto:jwm@proftpd.org) (4)
- [John Tunison <nosinut@cmu.edu>](mailto:nosinut@cmu.edu) (1)
- [Jonas Borgström <jonas@bengburken.net>](mailto:jonas@bengburken.net) (4)
- [Jonas Borgström <jonas_b@bitSMART.com>](mailto:jonas_b@bitSMART.com) (14)
- [Jonas Frantz](#) (2)
- [Jonathan Dale <jdale@fla.fujitsu.com>](mailto:jdale@fla.fujitsu.com) (12)
- [Jonathan Gardner <fr2asbury@yahoo.com>](mailto:fr2asbury@yahoo.com) (1)
- [Jonathan Gardner <gardnerj@provide.net>](mailto:gardnerj@provide.net) (11)
- [Jordan Ritter <jpr5@darkridge.com>](mailto:jpr5@darkridge.com) (6)

- [Joris van der Hoeven <vdhoeven@texmacs.org> \(2\)](#)
- [Jos Dehaes <jos.dehaes@bigfoot.com> \(6\)](#)
- [Jose M. Benitez <J.M.Benitez@decsai.ugr.es> \(2\)](#)
- [Josh Parsons <josh@philosophy.org.au> \(5\)](#)
- [Joshua Swink <jswink@pacbell.net> \(3\)](#)
- [José Romildo Malaquias <malaquias@iceb.ufop.br> \(4\)](#)
- [José Romildo Malaquias <romildo@iceb.ufop.br> \(25\)](#)
- [Juan Luis Baptiste <jbaptist@merlinux.org> \(2\)](#)
- [Juan Quintela <quintela@mandrakesoft.com> \(84\)](#)
- [Julian Missig <julian@jabber.org> \(17\)](#)
- [Jun Nishii <jun@vinelinux.org> \(4\)](#)
- [Justin Cormack <jpcl@doc.ic.ac.uk> \(7\)](#)
- [Justin Vallon <vallon@mindspring.com> \(4\)](#)
- [Jürgen Vigna <jug@sad.it> \(1\)](#)
- [KDE Packaging Team <kde-packager@kde.org> \(17\)](#)
- [KDE Packaging Team <redhat-rpms@kde.org> \(1\)](#)
- [KaYue Mak <gtkzip@yahoo.com> \(2\)](#)
- [Kai Vehmanen <kaiv@wakkanet.fi> \(30\)](#)
- [Kain <csa7fff@yahoo.com> \(2\)](#)
- [Kalle Dalheimer <kalle@kde.org> \(2\)](#)
- [Karl DeBisschop <karl@debisschop.net> \(6\)](#)
- [Karl DeBisschop <kdebisschop@users.sourceforge.net> \(18\)](#)
- [Karl Diedrich <died9501@uidaho.edu> \(10\)](#)
- [Karl Diedrich <ktdiedrich@yahoo.com> \(3\)](#)
- [Karl Grindley <karl.grindley@matchmaker.com> \(2\)](#)
- [Karl O. Pinc <kop@meme.com> \(4\)](#)
- [Karl Robillard <krobillard@home.com> \(1\)](#)
- [Karl Söderström coldn@gamearena.net \(2\)](#)
- [Karl-Martin Skontor <kskontor@cs.strath.ac.uk> \(3\)](#)
- [Karl-Martin Skontorp <karl-ms@online.no> \(3\)](#)
- [Karl-Martin Skontorp <kms@users.sourceforge.net> \(10\)](#)
- [Karol Pietrzak <noodlez84@earthlink.net> \(3\)](#)
- [Karsten Weiss <karsten@addx.au.s.shuttle.de> \(2\)](#)
- [Katherine Lim <kathil@melbpc.org.au> \(4\)](#)
- [Kayvan A. Sylvan <kayvan@sylvan.com> \(15\)](#)
- [Kazu Hirata <kazu@hxi.com> \(14\)](#)
- [Kazu Hirata <kban@hdl.co.jp> \(8\)](#)
- [Kazu Hirata <kban@hxi.com> \(6\)](#)
- [Keiichi Takahashi <bitwalk@jcom.home.ne.jp> \(2\)](#)
- [Keita Maehara <maehara@debian.org> and Christophe Bernard <bernard@cmapx.polytechnique.fr> \(4\)](#)
- [Keith Amidon <camalot@picnicpark.org> \(4\)](#)
- [Keith Clayton \(kclayton@jps.net\) \(11\)](#)
- [Keith Wesolowski <wesolows@cs.unr.edu> \(2\)](#)
- [Kelly Roestel <kelly@dolphinsearch.com> \(2\)](#)
- [Kenneth Porter <shiva@well.com> \(6\)](#)
- [Kenny Lim <kenny@predawnia.org> \(3\)](#)
- [Kensuke Iwahashi <ken@aqua.chem.nagoya-u.ac.jp> \(2\)](#)
- [Kevin I-Sen Lai <laik@cs.stanford.edu> \(10\)](#)
- [Kevin Vajk <kvajk@ricochet.net> \(1\)](#)
- [Kim Woelders <kim@woelders.dk> \(12\)](#)

- [Kirk Bauer <kirk@kaybee.org>](mailto:kirk@kaybee.org) (11)
- [Kitya Karlson <karlson@ncport.ru>](mailto:karlson@ncport.ru) (1)
- [Kjetil Wiekhorst Jørgensen <jorgens+rpm@pvv.org>](mailto:jorgens+rpm@pvv.org) (3)
- [Kjetil Wiekhorst Jørgensen <jorgens@fastfire.pvv.org>](mailto:jorgens@fastfire.pvv.org) (3)
- [Kjetil Wiekhorst Jørgensen <jorgens+rpm@pvv.org>](mailto:jorgens+rpm@pvv.org) (39)
- [Kjetil Wiekhorst Jørgensen <jorgens@pvv.org>](mailto:jorgens@pvv.org) (23)
- [Kjetil Wiekhorst Jørgensen <jorgens@zarhan.pvv.org>](mailto:jorgens@zarhan.pvv.org) (21)
- [Kjetil Wiekhorst Jørgensen, <jorgens+rpm@pvv.org>](mailto:jorgens+rpm@pvv.org) (4)
- [Kjetil Wiekhorst Jørgensen, <jorgens@zarhan.pvv.org>](mailto:jorgens@zarhan.pvv.org) (1)
- [Knut Anders Hatlen <kahatlen@online.no>](mailto:kahatlen@online.no) (2)
- [Konrad Hinsén <hinsen@cnrs-orleans.fr>](mailto:hinsen@cnrs-orleans.fr) (41)
- [Konstantin L. Metlov <metlov@fzu.cz>](mailto:metlov@fzu.cz) (2)
- [Konstanty Bialkowski](#) (6)
- [Koudijs <fulxo@users.sourceforge.net>](mailto:fulxo@users.sourceforge.net) (6)
- [Koudijs <koudijs@users.sourceforge.net>](mailto:koudijs@users.sourceforge.net) (21)
- [Kriang Lerdsuwanakij <lerdsuwa@users.sourceforge.net>](mailto:lerdsuwa@users.sourceforge.net) (88)
- [Lamar Owen <lamar.owen@wgcr.org>](mailto:lamar.owen@wgcr.org) (48)
- [Lars Kellogg-Stedman <lars@larsshack.org>](mailto:lars@larsshack.org) (2)
- [Laszlo Nemeth <nemeth@mzperx.ddns.org>](mailto:nemeth@mzperx.ddns.org) (2)
- [Laszlo Nemeth <nemeth@unixnet.org>](mailto:nemeth@unixnet.org) (1)
- [Laurent Cavallin \[lcavalli@ens-lyon.fr\] <lcavalli@ens-lyon.fr>](mailto:lcavalli@ens-lyon.fr) (2)
- [Laurent Cheylus <foxy@free.fr>](mailto:foxy@free.fr) (1)
- [Laurent Culioli <laurent@mandrakesoft.com>](mailto:laurent@mandrakesoft.com) (18)
- [Laurent Culioli <laurent@pschit.net>](mailto:laurent@pschit.net) (129)
- [Lauri Nurmi <lanurmi@iki.fi>](mailto:lanurmi@iki.fi) (1)
- [Lauris Kaplinski <lauris@ariman.ee>](mailto:lauris@ariman.ee) (10)
- [Len Budney](#) (3)
- [Lenny Cartier <l.cartier@gmx.net>](mailto:l.cartier@gmx.net) (14)
- [Lenny Cartier <lenny@mandrakesoft.com>](mailto:lenny@mandrakesoft.com) (3694)
- [Leo Milano <lmilano@udel.edu>](mailto:lmilano@udel.edu) (6)
- [Leonid Khramov](#) (2)
- [LessTif Releasemeister <lesstif@hungry.com>](mailto:lesstif@hungry.com) (88)
- [Lev Babiev <lev@mindless.com>](mailto:lev@mindless.com) (2)
- [Lev Makhliis <miev@writeme.com>](mailto:miev@writeme.com) (6)
- [Levente Farkas <lfarkas@inf.u-szeged.hu>](mailto:lfarkas@inf.u-szeged.hu) (12)
- [Levente Farkas <lfarkas@mindmaker.hu>](mailto:lfarkas@mindmaker.hu) (2)
- [Levente Farkas <lfarkas@ohsh.u-szeged.hu>](mailto:lfarkas@ohsh.u-szeged.hu) (41)
- [Levi Ramsey <levi@cygnetnet.net>](mailto:levi@cygnetnet.net) (16)
- [Levi Ramsey <leviramsey@linux-mandrake.com>](mailto:leviramsey@linux-mandrake.com) (2)
- [Linux-Mandrake Security Team <security@linux-mandrake.com>](mailto:security@linux-mandrake.com) (1129)
- [Linux-Mandrake Team <bugs@linux-mandrake.com>](mailto:bugs@linux-mandrake.com) (12)
- [Loban Rahman <loban@earthling.net>](mailto:loban@earthling.net) (7)
- [Lubos Lunak <l.lunak@email.cz>](mailto:l.lunak@email.cz) (1)
- [Luca Berra <bluca@vodka.it>](mailto:bluca@vodka.it) (28)
- [Lucian Langa <cooly@gyl.ro>](mailto:cooly@gyl.ro) (22)
- [Luciano Notarfrancesco <luciano@core-sdi.com>](mailto:luciano@core-sdi.com) (1)
- [Luciano Notarfrancesco <luciano@corest.com>](mailto:luciano@corest.com) (3)
- [Lumir Vanek <lvanek@eanet.cz>](mailto:lvanek@eanet.cz) (4)
- [Lyle Hanson \(lhanson@cs.nmu.edu\)](mailto:lhanson@cs.nmu.edu) (8)
- [Lyle Kempler <term@kempler.net>](mailto:term@kempler.net), [Joakim Bodin <bodin@dreamhosted.com>](mailto:bodin@dreamhosted.com) (67)
- [Lyle Kempler <term@twistedpath.org>](mailto:term@twistedpath.org), [Joakim Bodin <bodin@dreamhosted.com>](mailto:bodin@dreamhosted.com) (52)

- [MATSUURA Takanori <t-matsuu@protein.osaka-u.ac.jp>](mailto:t-matsuu@protein.osaka-u.ac.jp) (11)
- [MURANAKA Masaki <monaka@asahi-net.email.ne.jp>](mailto:monaka@asahi-net.email.ne.jp) (41)
- [Maciej Lesniewski <nimir@kis.p.lodz.pl>](mailto:nimir@kis.p.lodz.pl) (5)
- [Maksim V. Nikulin <nikulini@msiu.ru>](mailto:nikulini@msiu.ru) (2)
- [Mandrake <mandrake@valinux.com>](mailto:mandrake@valinux.com) (6)
- [Mandrake Linux KDE Team <kde@mandrakesoft.com>](mailto:kde@mandrakesoft.com) (598)
- [Mandrake Linux Team <bugs@linux-mandrake.com>](mailto:bugs@linux-mandrake.com) (9)
- [Mandrake Linux Team <http://www.mandrakeexpert.com>](http://www.mandrakeexpert.com) (3300)
- [Mandrake Linux Team <https://qa.mandrakesoft.com/>](https://qa.mandrakesoft.com/) (18)
- [Manfred H. Winter <mahowi@gmx.net>](mailto:mahowi@gmx.net) (2)
- [Manfred H. Winter <mahowi@gmx.net>](mailto:mahowi@gmx.net), [Christoph Lorenz <Christoph_Lorenz@gmx.de>](mailto:Christoph_Lorenz@gmx.de) (108)
- [Mano <mano78@users.sourceforge.net>](mailto:mano78@users.sourceforge.net) (33)
- [Manoj Kasichainula <manojk+rpm@io.com>](mailto:manojk+rpm@io.com) (9)
- [Manoj Kasichainula <manojk@io.com>](mailto:manojk@io.com) (9)
- [Manuel Amador \(Rudd-O\) <amadorm@usm.edu.ec>](mailto:amadorm@usm.edu.ec) (7)
- [Manuel McLure <mmclure@sac.verio.net>](mailto:mmclure@sac.verio.net) (4)
- [Manuel Menal <mmenal@netcourrier.com>](mailto:mmenal@netcourrier.com) (4)
- [Manuel Sugawara <masm@deprof.fciencias.unam.mx>](mailto:masm@deprof.fciencias.unam.mx) (1)
- [Marc André Selig <selig@users.sourceforge.net>](mailto:selig@users.sourceforge.net) (2)
- [Marc Ewing <marc@redhat.com>](mailto:marc@redhat.com) (83)
- [Marc Jauvin <marc@visualocity.com>](mailto:marc@visualocity.com) (2)
- [Marc Jauvin <mjauvin@email.com>](mailto:mjauvin@email.com) (2)
- [Marc Lavallée <odradek@videotron.ca>](mailto:odradek@videotron.ca) (3)
- [Marc Maurer <j.m.maurer@student.utwente.nl>](mailto:j.m.maurer@student.utwente.nl) (29)
- [Marc Merlin <marcsoft@magic.metawire.com>](mailto:marcsoft@magic.metawire.com) (3)
- [Marc Nozell <marc@nozell.com>](mailto:marc@nozell.com) (56)
- [Marc aka Uwog \[j.m.maurer@student.utwente.nl\]](mailto:j.m.maurer@student.utwente.nl) (3)
- [Marcel Pol <mpol@gmx.net>](mailto:mpol@gmx.net) (432)
- [Marcel Pol <mpol@mandrake.org>](mailto:mpol@mandrake.org) (70)
- [Marcelo Fernandes Vianna <m-vianna@usa.net>](mailto:m-vianna@usa.net) (8)
- [Marcin Woźniak <marcin@heban.com.pl>](mailto:marcin@heban.com.pl) (9)
- [Marco Ermini <linux@markoer.org>](mailto:linux@markoer.org) (11)
- [Marco Pesenti Gritti <mpeseng@tin.it>](mailto:mpeseng@tin.it) (216)
- [Marek Habersack <grendel@hoth.amu.edu.pl>](mailto:grendel@hoth.amu.edu.pl) (3)
- [Marek Habersack <grendel@vip.maestro.com.pl>](mailto:grendel@vip.maestro.com.pl) (3)
- [Marius Feraru <altblue@n0i.net>](mailto:altblue@n0i.net) (4)
- [Mark A. Wicks <mwicks@kettering.edu>](mailto:mwicks@kettering.edu) (2)
- [Mark Bainter <mark-e@cymry.org>](mailto:mark-e@cymry.org) (8)
- [Mark Bergsma <mark@mbergsma.demon.nl>](mailto:mark@mbergsma.demon.nl) (12)
- [Mark Biggers <biggers@ivc.com>](mailto:biggers@ivc.com) (1)
- [Mark Doliner <thekingant@users.sourceforge.net>](mailto:thekingant@users.sourceforge.net) (2)
- [Mark Horning <rip6@rip6.net>](mailto:rip6@rip6.net) (16)
- [Mark Knox <mark.knox@rebel.com>](mailto:mark.knox@rebel.com) (1)
- [Mark Lamourine](#) (39)
- [Mark Mitchell <mark@codesourcery.com>](mailto:mark@codesourcery.com) (2)
- [Mark Mitchell <mark@marmkitchell.com>](mailto:mark@marmkitchell.com) (2)
- [Mark Mitchell <mmitchell@usa.net>](mailto:mmitchell@usa.net) (2)
- [Mark Spencer <markster@marko.net>](mailto:markster@marko.net) (2)
- [Mark Turner <markt@indelible-blue.com>](mailto:markt@indelible-blue.com) (2)
- [Marko Grönroos <magi@iki.fi>](mailto:magi@iki.fi) (4)
- [Marko Macek <Marko.Macek@gmx.net>](mailto:Marko.Macek@gmx.net) (59)

- [Martin Baulig <martin@home-of-linux.org>](mailto:martin@home-of-linux.org) (22)
- [Martin Blais <blais@iro.umontreal.ca>](mailto:blais@iro.umontreal.ca) (31)
- [Martin Elsmann \(mael@dina.kvl.dk\)](mailto:mael@dina.kvl.dk) (2)
- [Martin Hamilton <martinh@gnu.org>](mailto:martinh@gnu.org) (2)
- [Martin Junius <mj@m-j-s.net>](mailto:mj@m-j-s.net) (8)
- [Martin Mevald <martinmv@penguin.cz>](mailto:martinmv@penguin.cz) (2)
- [Martin Norbäck <d95mback@dtek.chalmers.se>](mailto:d95mback@dtek.chalmers.se) (2)
- [Martin Preishuber <Martin.Preishuber@eclipt.uni-klu.ac.at>](mailto:Martin.Preishuber@eclipt.uni-klu.ac.at) (54)
- [Martin Schimschak <masch@theo-phys.uni-essen.de>](mailto:masch@theo-phys.uni-essen.de) (2)
- [Martin Seviior <mseviior@physics.unimelb.edu.au>](mailto:mseviior@physics.unimelb.edu.au) (16)
- [Martyn Plummer <plummer@iarc.fr>](mailto:plummer@iarc.fr) (172)
- [Masahiko Nakayado <masahiko@iname.com>](mailto:masahiko@iname.com) (2)
- [Matej Cepl <ceplma00@yahoo.com>](mailto:ceplma00@yahoo.com) (5)
- [MathForum RPM <rpm@mathforum.com>](mailto:rpm@mathforum.com) (2)
- [Mathias Hasselmann <massel@users.sourceforge.net>](mailto:massel@users.sourceforge.net) (6)
- [Mathias Hasselmann <mathias.hasselmann@gmx.de>](mailto:mathias.hasselmann@gmx.de) (66)
- [Matt Gauthier <elleron@yahoo.com>](mailto:elleron@yahoo.com) (8)
- [Matt Herbert <mherbert@vitts.net>](mailto:mherbert@vitts.net) (3)
- [Matt Shibla <mshibla@iname.com>](mailto:mshibla@iname.com) (4)
- [Matt Thompson <thompsma@colorado.edu>](mailto:thompsma@colorado.edu) (1)
- [Matt Wagner <matt@mysql.com>](mailto:matt@mysql.com) (12)
- [Matt Wagner <mwagner@mysql.com>](mailto:mwagner@mysql.com) (4)
- [Matteo Catanese <catanese@athena.polito.it>](mailto:catanese@athena.polito.it) (4)
- [Matthew Campbell <mattcamp@crosswinds.net>](mailto:mattcamp@crosswinds.net) (4)
- [Matthew Campbell <mattcamp@feist.com>](mailto:mattcamp@feist.com) (11)
- [Matthew Campbell <mattcampbell@pobox.com>](mailto:mattcampbell@pobox.com) (2)
- [Matthew Grant](#) (3)
- [Matthew J. Saltzman <mjs@clemson.edu>](mailto:mjs@clemson.edu) (4)
- [Matthew Miller <mattdm@mattdm.org>](mailto:mattdm@mattdm.org) (6)
- [Matthew Peterson <mpeterson@caldera.com>](mailto:mpeterson@caldera.com) (2)
- [Matthew Peterson <mpeterson@calderasystems.com>](mailto:mpeterson@calderasystems.com) (46)
- [Matthew R. MacIntyre <mrmacint@chat.carleton.ca>](mailto:mrmacint@chat.carleton.ca) (2)
- [Matthew Simpson <matthewsimpson@users.sourceforge.net>](mailto:matthewsimpson@users.sourceforge.net) (2)
- [Matthias Düsterhoeft <duesti@gmx.de>](mailto:duesti@gmx.de) (98)
- [Matthias Hensler <matthias@wspse.de>](mailto:matthias@wspse.de) (21)
- [Matthias Saou <matthias.saou@est.une.marmotte.net>](mailto:matthias.saou@est.une.marmotte.net) (1995)
- [Matthias Welwarsky <matze@stud.fbi.fh-darmstadt.de>](mailto:matze@stud.fbi.fh-darmstadt.de) (1)
- [Mattias Düsterhoeft <duesti@gmx.de>](mailto:duesti@gmx.de) (12)
- [Mattias Lindblad <matli@lysator.liu.se>](mailto:matli@lysator.liu.se) (1)
- [Maxim Krasnyansky <max_mk@yahoo.com>](mailto:max_mk@yahoo.com) (2)
- [MediKat Europe <support@medikat.com>](mailto:support@medikat.com) (5)
- [Merlin Hughes <merlin@merlin.org>](mailto:merlin@merlin.org) (4)
- [Michael Ashton <data@fleetsmart.com>](mailto:data@fleetsmart.com) (3)
- [Michael Ashton <data@users.sourceforge.net>](mailto:data@users.sourceforge.net) (7)
- [Michael Devogelaere <hoegie@digibel.be>](mailto:hoegie@digibel.be) (1)
- [Michael Edwardes <mte@users.sourceforge.net>](mailto:mte@users.sourceforge.net) (6)
- [Michael Fulbright <msf@redhat.com>](mailto:msf@redhat.com) (14)
- [Michael Goffioul <goffioul@emic.ucl.ac.be>](mailto:goffioul@emic.ucl.ac.be) (12)
- [Michael H. Schimek <mschimek@users.sourceforge.net>](mailto:mschimek@users.sourceforge.net) (5)
- [Michael Haggerty <mhagger@alum.mit.edu>](mailto:mhagger@alum.mit.edu) (2)
- [Michael Hauf <mh@hauf.net>](mailto:mh@hauf.net) (5)

- [Michael Hauf <mhauf@ansbach.netsurf.de> \(2\)](#)
- [Michael Jennings <mej@eterm.org> \(5\)](#)
- [Michael Jennings <mej@valinux.com> \(15\)](#)
- [Michael McLagan <michael.mclagan@linux.org> \(82\)](#)
- [Michael Nolta <mrnolta@users.sourceforge.net> \(31\)](#)
- [Michael Reinsch <mr@uue.org> \(4\)](#)
- [Michael Scherer <misc@mandrake.org> \(27\)](#)
- [Michael Scherer <scherer.michael@free.fr> \(235\)](#)
- [Michael Stefaniuc <mstefani@redhat.com> \(2\)](#)
- [Michael Still \(12\)](#)
- [Michael Stroeder <michael@stroeder.com> \(2\)](#)
- [Michael Sweet <mike@easysw.com> \(11\)](#)
- [Michael Wand <mwand@gmx.de> \(1\)](#)
- [Michal Choura <xchoura@fel.cvut.cz> \(2\)](#)
- [Michal Jaegermann <michal@ellpsapce.math.ualberta.ca> \(1\)](#)
- [Michal Svec <pm@penguin.cz> \(4\)](#)
- [Michal Svec <rebel@atrey.karlin.mff.cuni.cz> \(1\)](#)
- [Michal Svec <rebel@penguin.cz> \(2\)](#)
- [Michèl Alexandre Salim <aeruscator@yahoo.fr> \(4\)](#)
- [MieTerra LLC <bugs@mieterra.com> \(2\)](#)
- [Miguel Armas <kuko@ulpgc.es> \(2\)](#)
- [Mihai Bazon <mishoo@infoiasi.ro> \(13\)](#)
- [Mihai Ibanescu <misa@dntis.ro> \(5\)](#)
- [Mihai Ibanescu <misa@necomm.ro> \(15\)](#)
- [Mika Lindqvist <mika@field.ttaol.fi> \(3\)](#)
- [Mike Jagdis <jaggy@purplet.demon.co.uk> \(2\)](#)
- [Mike Newman <packages@principia-systems.co.uk> \(12\)](#)
- [Mike Nolta <mrnolta@users.sourceforge.net> \(4\)](#)
- [Mike Oliphant <oliphant@gtk.org> \(69\)](#)
- [Mike Oliphant <oliphant@ling.ed.ac.uk> \(6\)](#)
- [Mike Ricketts \(1\)](#)
- [Mike Ricketts <rickettm@ox.compsoc.net> \(2\)](#)
- [Mike Wyer <m.wyer@ic.ac.uk> \(2\)](#)
- [Mikkel L. Ellertson <mikkel@execpc.com> \(5\)](#)
- [Miles Egan <miles@caddr.com> \(2\)](#)
- [Miles Lott <milos@insync.net> \(4\)](#)
- [Mirko Zeibig <mirko@zeibig.net> \(8\)](#)
- [Mirko Zeibig <mz@webideal.de> \(14\)](#)
- [Montel Laurent <lmontel@mandrakesoft.com> \(1203\)](#)
- [Moonshine Development Team <moonshine@suite3220.com> \(2\)](#)
- [Mundi King <kingmundi@hotmail.com> \(4\)](#)
- [Murray Todd Williams <MurrayTodd@yahoo.com> \(16\)](#)
- [Naba kumar <kh_naba@users.sourceforge.net> \(4\)](#)
- [Naba kumar <kh_naba@yahoo.com> \(9\)](#)
- [NanarDon <nanardon@mandrake.org> \(30\)](#)
- [Nassib Nassar <nassar@etymon.com> \(1\)](#)
- [Nathan 'Nato' Uno \(1\)](#)
- [Nathan Hand <h9304891@student.anu.edu.au> \(3\)](#)
- [Nathan Hjelm <hjelm@unm.edu> \(4\)](#)
- [Nathan Hjelm <hjelm@users.sourceforge.net> \(12\)](#)
- [Neil Gorsuch <njg@pdnt.com> \(4\)](#)

- [Neil Winton \(neil@winton.org.uk\)](mailto:neil@winton.org.uk) (2)
- [NetWinder Autobuild <build@netwinder.org>](mailto:build@netwinder.org) (9)
- [Nguyen-Dai Quy <NDQ@iris.ltas.ulg.ac.be>](mailto:NDQ@iris.ltas.ulg.ac.be) (2)
- [Nic Roets \(glpoisson@sig.co.za\)](mailto:glpoisson@sig.co.za) (2)
- [Nick Betcher <nbetcher@real-time.com>](mailto:nbetcher@real-time.com) (3)
- [Nick Urbanik <nicku@iohk.com>](mailto:nicku@iohk.com) (2)
- [Nick Urbanik <nicku@vtc.edu.hk>](mailto:nicku@vtc.edu.hk) (23)
- [Nicolas M. Thiery <nthiery@mines.edu>](mailto:nthiery@mines.edu) (10)
- [Nicolas M. Thiery <nthiery@users.sourceforge.net>](mailto:nthiery@users.sourceforge.net) (11)
- [Nicolas Mailhot <Nicolas.Mailhot \(at\) JPackage.org>](mailto:Nicolas.Mailhot (at) JPackage.org) (15)
- [Nicolas Mailhot <Nicolas.Mailhot at laPoste.net>](mailto:Nicolas.Mailhot at laPoste.net) (20)
- [Nicolas Mailhot <Nicolas.Mailhot@email.enst.fr>](mailto:Nicolas.Mailhot@email.enst.fr) (6)
- [Nicolas Planel <nplanel@mandrakesoft.com>](mailto:nplanel@mandrakesoft.com) (20)
- [Nicolas Rougier <Nicolas.Rougier@loria.fr>](mailto:Nicolas.Rougier@loria.fr) (4)
- [Nicolas Thiéry «Isil» \(Nicolas.Thiery@ens.fr\)](mailto:Nicolas.Thiery@ens.fr) 1998 (2)
- [Nicolas Vignal <nicolas.vignal@fnac.net>](mailto:nicolas.vignal@fnac.net) (16)
- [Niels Reedijk](#) (32)
- [Niemeyer Network <packages@niemeyer.net>](mailto:packages@niemeyer.net) (2)
- [Nils Philippsen <n timer@rhlx01.rz.fht-esslingen.de>](mailto:nils@rhlx01.rz.fht-esslingen.de) (3)
- [ObiTuarY <Obituary@cybernet.be>](mailto:Obituary@cybernet.be) (40)
- [ObiTuarY <obituary@freshmeat.net>](mailto:obituary@freshmeat.net) (343)
- [ObiTuarY <obituary@fresmeat.net>](mailto:obituary@fresmeat.net) (2)
- [Oden Eriksson <oden.eriksson@linux-mandrake.com>](mailto:oden.eriksson@linux-mandrake.com) (2138)
- [Oerjan Nygaard Austvold <orjana@acm.org>](mailto:orjana@acm.org) (2)
- [Okki <crow@planete.net>](mailto:crow@planete.net) (46)
- [Okki <okki@wanadoo.fr>](mailto:okki@wanadoo.fr) (21)
- [Ole Craig <www.cs.umass.edu/~olc>](http://www.cs.umass.edu/~olc) (2)
- [Oliver Schulze L. <oliver@pla.net.py>](mailto:oliver@pla.net.py) (4)
- [Olivier Blin <blino@mandrake.org>](mailto:blino@mandrake.org) (2)
- [Olivier Blin <oliv.blin@laposte.net>](mailto:oliv.blin@laposte.net) (81)
- [Olivier Fourdan <fourdan@xfce.org>](mailto:fourdan@xfce.org) (43)
- [Olivier Thauvin <thauvin@aerov.jussieu.fr>](mailto:thauvin@aerov.jussieu.fr) (843)
- [Orjan Nygaard Austvold <austvold@acm.org>](mailto:austvold@acm.org) (2)
- [Ottavio G. Rizzo <rizzo@pluto.linux.it>](mailto:rizzo@pluto.linux.it) (2)
- [Ousmane Wilane <wilane@mint.sn>](mailto:wilane@mint.sn) (2)
- [Owen Taylor <otaylor@redhat.com>](mailto:otaylor@redhat.com) (6)
- [PLD bug tracking system \(http://bugs.pld.org.pl/ \)](http://bugs.pld.org.pl/) (24441)
- [Pablo Costa <pablo@ib.usp.br>](mailto:pablo@ib.usp.br) (40)
- [Pablo Costa <pablo@shark.ib.usp.br>](mailto:pablo@shark.ib.usp.br) (2)
- [Pablo Saratxaga <pablo@mandrakesoft.com>](mailto:pablo@mandrakesoft.com) (432)
- [Pablo Saratxaga <srtxg@chanae.alphanet.ch>](mailto:srtxg@chanae.alphanet.ch) (80)
- [Packager: Philip Long <plong@mitre.org>](mailto:plong@mitre.org) (1)
- [Paolo Bacchilega <paolo.bacch@tin.it>](mailto:paolo.bacch@tin.it) (2)
- [Parnobis B.V. <packagers@parnobis.com>](mailto:packagers@parnobis.com) (10)
- [Pascal Bleser <guru@linuxbe.org>](mailto:guru@linuxbe.org) (1)
- [Pascal Robinet <pascal@sigann.net>](mailto:pascal@sigann.net) (1)
- [Pascal Terjan <CMoi@tuxfamily.org>](mailto:CMoi@tuxfamily.org) (75)
- [Pascal Terjan <pterjan.mandrake@org>](mailto:pterjan.mandrake@org) (26)
- [Pascal Terjan <pterjan@mandrake.org>](mailto:pterjan@mandrake.org) (10)
- [Pat Hennessy <path@magpage.com>](mailto:path@magpage.com) (6)
- [Patrick Charles <http://www.sf.net/users/pcharles>](http://www.sf.net/users/pcharles) (12)

- [Patrick Powell <papowell@astart.com>](mailto:papowell@astart.com) (8)
- [Patrik Hall <patrik.hall@pistetietoliikenne.com>](mailto:patrik.hall@pistetietoliikenne.com) (6)
- [Paul Bournival <cajun@current.nu>](mailto:cajun@current.nu) (3)
- [Paul Gear <paul@gear.dyndns.org>](mailto:paul@gear.dyndns.org) (14)
- [Paul Gear <pgear@redlands.qld.edu.au>](mailto:pgear@redlands.qld.edu.au) (3)
- [Paul Graunke <ptg@rice.edu>](mailto:ptg@rice.edu) (3)
- [Paul H. Hargrove <hargrove@sccm.Stanford.EDU>](mailto:hargrove@sccm.Stanford.EDU) (24)
- [Paul Marquis <pmarquis@pobox.com>](mailto:pmarquis@pobox.com) (3)
- [Paul Nasrat <pauln@truemesh.com>](mailto:pauln@truemesh.com) (46)
- [Paul Sheer <nospam@nospam.com>](mailto:nospam@nospam.com) (4)
- [Paul Sheer <psheer@obsidian.co.za>](mailto:psheer@obsidian.co.za) (4)
- [Paulo Matos <pjism@fct.unl.pt>](mailto:pjism@fct.unl.pt) (5)
- [Pavel Janik ml. <Pavel.Janik@linux.cz>](mailto:Pavel.Janik@linux.cz) (2)
- [Pavel Janik ml. <Pavel.Janik@inet.cz>](mailto:Pavel.Janik@inet.cz) (2)
- [Pawel Salek <pawsa@theochem.kth.se>](mailto:pawsa@theochem.kth.se) (1)
- [Pehr Anderson](mailto:pehr@alum.mit.edu) (1)
- [Pehr Anderson <pehr@alum.mit.edu>](mailto:pehr@alum.mit.edu) (1)
- [Pehr Anderson <pehr@pehr.net>](mailto:pehr@pehr.net) (2)
- [Per Øyvind Karlsen <peroyvind@linux-mandrake.com>](mailto:peroyvind@linux-mandrake.com) (865)
- [Per Øyvind Karlsen <peroyvind@sintrax.net>](mailto:peroyvind@sintrax.net) (1263)
- [Pete Ratzlaff <pratzlaff@cfa.harvard.edu>](mailto:pratzlaff@cfa.harvard.edu) (2)
- [Peter C. Norton <spacey@pobox.com>](mailto:spacey@pobox.com) (1)
- [Peter Chen <petechen@netilla.com>](mailto:petechen@netilla.com) (2)
- [Peter Csizmadia <cspeter@rmki.kfki.hu>](mailto:cspeter@rmki.kfki.hu) (2)
- [Peter Denison <peterd@pnd-pc.demon.co.uk>](mailto:peterd@pnd-pc.demon.co.uk) (2)
- [Peter Hanecak <hanecak@megaloman.com>](mailto:hanecak@megaloman.com) (2)
- [Peter Hanecak <hanecak@megaloman.sk>](mailto:hanecak@megaloman.sk) (96)
- [Peter Selinger <selinger@theory.stanford.edu>](mailto:selinger@theory.stanford.edu) (2)
- [Peter Soos <sp@osb.hu>](mailto:sp@osb.hu) (48)
- [Peter Vreman <peter@freepascal.org>](mailto:peter@freepascal.org) (2)
- [Petri Turunen <petri.turunen@raketti.net>](mailto:petri.turunen@raketti.net) (12)
- [Petri Turunen <petritur@freenet.hut.fi>](mailto:petritur@freenet.hut.fi) (11)
- [Phil Thompson <phil@river-bank.demon.co.uk>](mailto:phil@river-bank.demon.co.uk) (10)
- [Philip Long <plong@mitrc.org>](mailto:plong@mitrc.org) (30)
- [Philip Warren <pkw@duke.edu>](mailto:pkw@duke.edu) (10)
- [Philipp Gühring <p.guehring@poboxes.com>](mailto:p.guehring@poboxes.com) (1)
- [Phillip K. Hornung <falcon@pknet.com>](mailto:falcon@pknet.com) (1)
- [Pier Luigi Fiorini <plfiorini@users.sourceforge.net>](mailto:plfiorini@users.sourceforge.net) (4)
- [Pierre Scotney <pierre@mail.medstv.unimelb.edu.au>](mailto:pierre@mail.medstv.unimelb.edu.au) (48)
- [Piotr Habeko <gumis@wsi.edu.pl>](mailto:gumis@wsi.edu.pl) (2)
- [Piotr Majka <charvel@Amnesiac.mtl.pl>](mailto:charvel@Amnesiac.mtl.pl) (1)
- [Pixel <pixel@mandrakesoft.com>](mailto:pixel@mandrakesoft.com) (189)
- [Planar <Damien.Doligez@inria.fr>](mailto:Damien.Doligez@inria.fr) (5)
- [Planel Nicolas <nplanel@mandrakesoft.com>](mailto:nplanel@mandrakesoft.com) (76)
- [Prana <pranalukas@linux-mandrake.com>](mailto:pranalukas@linux-mandrake.com) (2)
- [Pranesh Ramjith <pranesh@beth.uniforum.org.za>](mailto:pranesh@beth.uniforum.org.za) (2)
- [Preston Hunt <preston@techwood.org>](mailto:preston@techwood.org) (2)
- [Progfou <Jean-Christophe.ANDRE@aupelf.refer.org>](mailto:Jean-Christophe.ANDRE@aupelf.refer.org) (4)
- [R Shapiro <reshapiro@mediaone.net>](mailto:reshapiro@mediaone.net) (5)
- [RPM Account <rpbuilder@rhodesmill.org>](mailto:rpbuilder@rhodesmill.org) (4)
- RPMForge.net (3)

- [Radek Liboska <liboska@uochb.cas.cz>](mailto:liboska@uochb.cas.cz) (2)
- [Radey Shouman <shouman@ne.mediaone.net>](mailto:shouman@ne.mediaone.net) (4)
- [Rajko Albrecht <rajko.albrecht@informatik.tu-chemnitz.de>](mailto:rajko.albrecht@informatik.tu-chemnitz.de) (11)
- [Rajko Albrecht <ral@alwins-world.de>](mailto:ral@alwins-world.de) (3)
- [Ralph Siemsen <ralphs@netwinder.org>](mailto:ralphs@netwinder.org) (8)
- [Ralph Slooten <axllent@axllent.cjb.net>](mailto:axllent@axllent.cjb.net) (2)
- [Ramin Charles Nakisa <ramin.nakisa@ncr.com>](mailto:ramin.nakisa@ncr.com) (2)
- [Ramiro Estrugo <ramiro@netscape.com>](mailto:ramiro@netscape.com) (3)
- [Ramiro Morales <rmrpms@usa.net>](mailto:rmrpms@usa.net) (2)
- Rampant Solutions, LLC (2)
- [Ramón García <ramon@juguete.quim.ucm.es>](mailto:ramon@juguete.quim.ucm.es) (2)
- [Randy Oyarzabal <techno91@users.sourceforge.net>](mailto:techno91@users.sourceforge.net) (3)
- [Raymond H. Kraft <ray@eskimo.com>](mailto:ray@eskimo.com) (2)
- [Raymond Wu <rwu@ml.com>](mailto:rwu@ml.com) (1)
- [Real Time Enterprises, Inc. <bugs@real-time.com>](mailto:bugs@real-time.com) (23)
- [Real Time Enterprises, Inc. <rpmbugs@real-time.com>](mailto:rpmbugs@real-time.com) (4)
- [Real Time Enterprises, Inc. <support@real-time.com>](mailto:support@real-time.com) (17)
- [RebuM <rebum@gmx.at>](mailto:rebum@gmx.at) (8)
- [Red Hat Contrib-Net <rhcnc-bugs@redhat.com>](mailto:rhcnc-bugs@redhat.com) (24)
- [Red Hat ContribNet <rhcnc-bugs@redhat.com>](mailto:rhcnc-bugs@redhat.com) (135)
- [Red Hat Software <bugs@redhat.com>](mailto:bugs@redhat.com) (1989)
- [Red Hat Software <http://developer.redhat.com/bugzilla/>](http://developer.redhat.com/bugzilla/) (577)
- [Red Hat Software <http://developer.redhat.com/bugzilla>](http://developer.redhat.com/bugzilla) (1628)
- [Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>](http://bugzilla.redhat.com/bugzilla) (21168)
- [Red Hat, Inc. <http://developer.redhat.com/bugzilla>](http://developer.redhat.com/bugzilla) (821)
- [Reinhard Katzmann <reinhard@suamor.de>](mailto:reinhard@suamor.de) (4)
- [Reinhold Schoeb <Reinhold.Schoeb@pcd.daimler-benz.com>](mailto:Reinhold.Schoeb@pcd.daimler-benz.com) (1)
- [Reinhold Schoeb <schoeba@str.daimler-benz.com>](mailto:schoeba@str.daimler-benz.com) (2)
- [Renaud Chaillat <rchaillat@mandrakesoft.com>](mailto:rchaillat@mandrakesoft.com) (4)
- [Renaud Chaillat <renaud_chaillat@netcourrier.com>](mailto:renaud_chaillat@netcourrier.com) (14)
- [Renaud Michel <renaud.michel@student.ulg.ac.be>](mailto:renaud.michel@student.ulg.ac.be) (10)
- [Revar Desmera <revar@belfry.com>](mailto:revar@belfry.com) (7)
- [Rex Dieter <rdieter@math.unl.edu>](mailto:rdieter@math.unl.edu) (3)
- [Rex Dieter <rdieter@unl.edu>](mailto:rdieter@unl.edu) (11)
- [Ricardo Bonon <bonon@users.sourceforge.net>](mailto:bonon@users.sourceforge.net) (1)
- [Ricardo Quesada <riq@core-sdi.com>](mailto:riq@core-sdi.com) (17)
- [Ricardo Quesada <riq@corest.com>](mailto:riq@corest.com) (3)
- [Rich Gade <rgade@users.sourceforge.net>](mailto:rgade@users.sourceforge.net) (28)
- [Richard Holcombe <raholcom@eos.ncsu.edu>](mailto:raholcom@eos.ncsu.edu) (4)
- [Richard Kinder <r_kinder@yahoo.com>](mailto:r_kinder@yahoo.com) (7)
- [Richard Torkar <ds98rito@thn.htu.se>](mailto:ds98rito@thn.htu.se) (27)
- [Richard van Hees <R.M.vanHees@fys.ruu.nl>](mailto:R.M.vanHees@fys.ruu.nl) (1)
- [Rickard Osser <ricky@osser.se>](mailto:ricky@osser.se) (1)
- [Riku Meskanen <mesrik@cc.jyu.fi>](mailto:mesrik@cc.jyu.fi) (11)
- [Riku Meskanen <mesrik@iki.fi>](mailto:mesrik@iki.fi) (2)
- [RiÄ?ardas Ä?epas <rch@WriteMe.Com>](mailto:rch@WriteMe.Com) (2)
- [Rob Clark <robert@cstr.ed.ac.uk>](mailto:robert@cstr.ed.ac.uk) (50)
- [Rob Flynn <rob@marko.net>](mailto:rob@marko.net) (228)
- [Rob Flynn <rob@tgflinux.com>](mailto:rob@tgflinux.com) (19)
- [Rob Lineweaver <rblin@wm.edu>](mailto:rblin@wm.edu) (6)
- [Rob McMillin <rlm@pricegrabber.com>](mailto:rlm@pricegrabber.com) (3)

- [Rob Riggs <rob@devilsthumb.com>](mailto:rob@devilsthumb.com) (3)
- [Robbie Ward <linuxphreak@gmx.co.uk>](mailto:linuxphreak@gmx.co.uk) (4)
- [Robbie Ward <robbie555@wardr23.fsnet.co.uk>](mailto:robbie555@wardr23.fsnet.co.uk) (8)
- [Robert "Jeff"rey Morriss <bugs@pobox.com>](mailto:bugs@pobox.com) (2)
- [Robert "Jeff"rey Morriss <morriss@pobox.com>](mailto:morriss@pobox.com) (4)
- [Robert B. Andrews <rba@InfoCorps.Net>](mailto:rba@InfoCorps.Net) (1)
- [Robert J. Tanner <tanner@real-time.com>](mailto:tanner@real-time.com) (48)
- [Robert Loomans <robert@netizen.com.au>](mailto:robert@netizen.com.au) (2)
- [Robert Loomans <robertl@premium.com.au>](mailto:robertl@premium.com.au) (3)
- [Robert Richard George 'reptile' Wal <reptile@cs.net.pl>](mailto:reptile@cs.net.pl) (8)
- [Robert S. Maier <rsm@math.arizona.edu>](mailto:rsm@math.arizona.edu) (5)
- [Robert Zilbauer <rpm@zilbauer.com>](mailto:rpm@zilbauer.com) (1)
- [Robert de Vries <R.de.Vries@fokkerspace.nl>](mailto:R.de.Vries@fokkerspace.nl) (2)
- [Roberto Alameda](mailto:roberto@myokay.net) (2)
- [Roberto Alameda <roberto@myokay.net>](mailto:roberto@myokay.net) (1)
- [Roberto Machorro <rmach@bigfoot.com>](mailto:rmach@bigfoot.com) (2)
- [Roberto Machorro <rmach@ciberlinux.net>](mailto:rmach@ciberlinux.net) (8)
- [Roberto Virga <rvirga@users.sourceforge.net>](mailto:rvirga@users.sourceforge.net) (1)
- [Robin Dunn <robin@alldunn.com>](mailto:robin@alldunn.com) (59)
- [Rod Cordova <rcordova@ethernet.org>](mailto:rcordova@ethernet.org) (2)
- [Rodrigo Barbosa <rodrigob@bh.conectiva.com.br>](mailto:rodrigob@bh.conectiva.com.br) (3)
- [Rodrigo Damazio <root@vros.com>](mailto:root@vros.com) (4)
- [Romain Lievin <rlievin@mail.com>](mailto:rlievin@mail.com) (3)
- [Ron Bickers <rbickers@logicetc.com>](mailto:rbickers@logicetc.com) (40)
- [Ron Brandner <r.brandner@teleweb.at>](mailto:r.brandner@teleweb.at) (2)
- [Ronan-Yann Lorin <lorin@adesium-services.fr>](mailto:lorin@adesium-services.fr) (7)
- [Ronan-Yann Lorin <lorin@mygale.org>](mailto:lorin@mygale.org) (6)
- [Ronny Haryanto <ronny@haryan.to>](mailto:ronny@haryan.to) (4)
- [Ross Arnold <ronz@users.sourceforge.net>](mailto:ronz@users.sourceforge.net) (2)
- [Rui M. Silva Seabra <rms@multicert.com>](mailto:rms@multicert.com) (3)
- [Russell Lang <http://www.cs.wisc.edu/~ghost/rjl.html>](http://www.cs.wisc.edu/~ghost/rjl.html) (2)
- [Russell Stuart <russell@stuart.wattle.id.au>](mailto:russell@stuart.wattle.id.au) (3)
- [Ruud Schrampp <r.schramp@kpn.com>](mailto:r.schramp@kpn.com) (2)
- [Ryan W. Maple <ryan@guardiandigital.com>](mailto:ryan@guardiandigital.com) (195)
- [Ryan Weaver <ryanw@falsehope.com>](mailto:ryanw@falsehope.com) (124)
- [Ryan Weaver <ryanw@infohwy.com>](mailto:ryanw@infohwy.com) (701)
- [Rémi Denis-Courmont <rdenis@simphalempin.com>](mailto:rdenis@simphalempin.com) (2)
- [SIS Devel Team <sisuite-devel@lists.sf.net>](mailto:sisuite-devel@lists.sf.net) (8)
- [Saadiq Rodgers-King <saadiqr@princeton.edu>](mailto:saadiqr@princeton.edu) (2)
- [Sam "Criswell" Hart <criswell@geekcomix.com>](mailto:criswell@geekcomix.com) (8)
- [Sam <slegros@mandrakesoft.com>](mailto:slegros@mandrakesoft.com) (1)
- [Sam Hart <criswell@geekcomix.com>](mailto:criswell@geekcomix.com) (1)
- [Sam Lantinga <hercules@lokigames.com>](mailto:hercules@lokigames.com) (10)
- [Sam Lantinga <slouken@devolution.com>](mailto:slouken@devolution.com) (6)
- [Sam Varshavchik <mrsam@geocities.com>](mailto:mrsam@geocities.com) (5)
- [Sander Pronk <pronk@amolf.nl>](mailto:pronk@amolf.nl) (2)
- [Sarod yatawatta <sarod@cs.pdn.ac.lk>](mailto:sarod@cs.pdn.ac.lk) (11)
- [Sayamindu Dasgupta <unmadindu@bengalinux.org>](mailto:unmadindu@bengalinux.org) (2)
- [Scott Haug <scott@id3.org>](mailto:scott@id3.org) (12)
- [Scott Lampert <fortunato@heavymetal.org>](mailto:fortunato@heavymetal.org) (9)
- [Scott Pakin <pakin@uiuc.edu>](mailto:pakin@uiuc.edu) (42)

- [Scott Sams <sbsams@digitallabyrinth.com> \(4\)](#)
- [Sean Dague \(4\)](#)
- [Sean Gabriel <gabriel@korsoft.com> \(4\)](#)
- [Sean P. Kane - kane@ca.metsci.com \(2\)](#)
- [Sean P. Kane <kane@ca.metsci.com> \(3\)](#)
- [Sean P. Kane <kanesp@evolveinc.net> \(2\)](#)
- [Sean Reifschneider <jafo-rpms@tummy.com> \(11\)](#)
- [Sean Summers <seans@acm.org> \(9\)](#)
- [Sebastian Frankfurt <sf@infesto.de> \(12\)](#)
- [Sebastian Schaffert <wastl@wastl.net> \(8\)](#)
- [Sebastiano Vigna <vigna at acm.org> \(20\)](#)
- [Sebastien Robin <seb@nexedi.com> \(88\)](#)
- [Serge Droz \(droz@physics.uoguelph.ca\) 1998 \(4\)](#)
- [Seth Vidal <skvidal@phy.duke.edu> \(2\)](#)
- [Sharif Nassar <sharif@ccs.ucsb.edu> \(2\)](#)
- [Silicon Graphics, Inc. <http://www.sgi.com/> \(33\)](#)
- [Silvio Schuerer <silvios@gmx.de> \(29\)](#)
- [Silvio Schuerer <silvios@gmx.net> \(9\)](#)
- [Simon Baldwin <simonb@caldera.com> \(4\)](#)
- [Simon Clift \(ssclift at cuic dot ca\) \(2\)](#)
- [Simon J Mudd <sjmudd@pobox.com> \(5\)](#)
- [Simon J Mudd <sjmudd@pobox.com> but see description below \(5\)](#)
- [Simon Matter <simix@datacomm.ch> \(2\)](#)
- [Simon Matter <simon.matter@invoca.ch> \(6\)](#)
- [Simon White <s_a_white@email.com> \(31\)](#)
- [Slava Pestov <slava@jedit.org> \(8\)](#)
- [Slava Pestov <sp@gjt.org> \(4\)](#)
- [Slavek Banko <slavek.banko@axis.cz> \(6\)](#)
- [Soenke J. Peters <peters+rpm@opcenter.de> \(2\)](#)
- [Soenke J. Peters <peters+rpm@simprovement.com> \(10\)](#)
- [Soenke J. Peters <peters@simprovement.com> \(1\)](#)
- [Soenke J. Peters <soenke+rpm@simprovement.com> \(2\)](#)
- [Soos Peter <sp@osb.hu> \(3\)](#)
- [Sorin Lingureanu <sorin@tuiasi.ro> \(6\)](#)
- [Spencer Anderson <sdander@oberon.ark.com> \(117\)](#)
- [Spiros Papadimitriou \(2\)](#)
- [Stefan Gartner <Stefan.Gartner@fhs-hagenberg.ac.at> \(2\)](#)
- [Stefan Hornburg <racke@gundel.han.de> \(4\)](#)
- [Stefan Ondrejicka <ondrej@idata.sk> \(30\)](#)
- [Stefan Suchi <suchi@gmx.de> \(6\)](#)
- [Stefan Waldherr <stefan@waldherr.org> \(3\)](#)
- [Stefan Waldherr <swa@cs.cmu.edu> \(5\)](#)
- [Stefane Fermigier <sf@nuxeo.com> \(1\)](#)
- [Stefano Tognon <ice00@users.sourceforge.net> \(15\)](#)
- [Steffan Olesen <steffan@mail1.stofanet.dk> \(1\)](#)
- [Steffen Sobiech <ssobiech@cityweb.de> \(2\)](#)
- [Stelian Pop <stelian@popies.net> \(32\)](#)
- [Stelios Dramis <sdramis@egnatia.ee.auth.gr> \(8\)](#)
- [Stephen Thomas <stephen@thomas.mitre.org> \(1\)](#)
- [Stephen White <swhite@ox.compsoc.net> \(2\)](#)
- [Stephen Williams <steve@icarus.com> \(3\)](#)

- [Steve Beattie <steve@wirex.net> \(5\)](#)
- [Steve Murray \(stevem@eng.uts.edu.au\) \(6\)](#)
- [Steven G. Johnson <stevenj@alum.mit.edu> \(2\)](#)
- [Steven Hatfield \(2\)](#)
- [Steven J Tucker <classics@nacs.net> \(2\)](#)
- [Steven J. Hill <sjhill@plutonium.net> \(5\)](#)
- [Stew Benedict <sbenedict@mandrakesoft.com> \(465\)](#)
- [Stéfane Fermigier, Nuxeo <sf@nuxeo.com> \(6\)](#)
- [Stéphane Genaud <genaud@icps.u-strasbg.fr> \(29\)](#)
- [Sumit Khanna <skhanna@csc.tntech.edu> \(11\)](#)
- [Sven Marth <smarth@marth.com> \(3\)](#)
- [Sydney Tang <stang@users.sourceforge.net> \(1\)](#)
- [Sydney Tang <sydney.tang@computer.org> \(6\)](#)
- [Sylvain GIL <sjog@tootella.org> \(1\)](#)
- [Sylvestre Taburet <staburet@mandrakesoft.com> \(14\)](#)
- [Szabolcs Seláf <selu@users.sourceforge.net> \(8\)](#)
- [T. V. Raman <raman@cs.cornell.edu> \(17\)](#)
- [TUCAN Devel Team <tucan@difinium.com> \(4\)](#)
- [TUCAN Devel Team <tucan@tgflinux.com> \(18\)](#)
- [Takeda Eiji <vine@flatout.org> \(10\)](#)
- [Term <kempler@utdallas.edu> \(13\)](#)
- [Tero Favorin <tero@favorin.com> \(3\)](#)
- [Terra Soft Solutions <http://bugs.yellowdoglinux.com> \(4737\)](#)
- [Terék Zsolt <tz124@hszk.bme.hu> \(2\)](#)
- [Tetsuhiro Nakane <nakane@ecc.denso.co.jp> \(2\)](#)
- [Tharin <k_becker@informatik.uni-kl.de> \(2\)](#)
- [The BerLinux <the.berlinux@berlinux.in-berlin.de> \(2\)](#)
- [The Meme Factory Inc. http://www.meme.com \(4\)](#)
- [The Mercury Group <mercury@cs.mu.oz.au> \(2\)](#)
- [The OpenPKG Project \(19\)](#)
- [The Rasterman <raster@rasterman.com> \(18\)](#)
- [The Rasterman <raster@rasterman.com> Lyle Kempler <term@kempler.net>, Joakim Bodin <bodin@dreamhosted.com> \(14\)](#)
- [The Rasterman <raster@rasterman.com> Term <term@kempler.net> \(6\)](#)
- [The Rasterman <raster@redhat.com> \(25\)](#)
- [The Rasterman <raster@redhat.com> \(3\)](#)
- [The pnm2ppa team <ppa-rpms@users.sourceforge.net> \(16\)](#)
- [The pnm2ppa team <ppa-rpms@users.sourceforge.net> \(11\)](#)
- [Theodore C. Belding <Ted.Belding@umich.edu> \(2\)](#)
- [Thibaut Cousin <linux@thibaut-cousin.net> \(14\)](#)
- [Thierry Godefroy <xdialog@free.fr> \(1\)](#)
- [Thierry Vignaud <tvignaud@mandrakesoft.com> \(1000\)](#)
- [Thierry khousakoun <tkhousakoun@mandrakesoft.com> \(4\)](#)
- [Thilo Mezger <thilo@aifb.uni-karlsruhe.de> \(3\)](#)
- [Thomas Backlund <tmb@iki.fi> \(8\)](#)
- [Thomas Backlund <tmb@mandrake.org> \(1\)](#)
- [Thomas Baumgart <ipwizard@users.sourceforge.net> \(1\)](#)
- [Thomas Capricelli <orzel@yalbi.com> \(2\)](#)
- [Thomas Hasart \(1\)](#)
- [Thomas Juerges <thomas.juerges@astro.ruhr-uni-bochum.de> \(2\)](#)
- [Thomas Leonard <tal197@users.sourceforge.net> \(2\)](#)

- [Thomas Leonard <tal197@users.sourceforge.net>, Andy Piper <andy.piper@freeuk.com> \(1\)](#)
- [Thomas Ribbrock <emgaron@gmx.net> \(16\)](#)
- [Thomas Tallberg <ttallber@cc.hut.fi> \(2\)](#)
- [Thornton Prime <thor@primenet.com> \(2\)](#)
- [Thorsten Kukuk <kukuk@vt.uni-paderborn.de> \(3\)](#)
- [Tibor Pittich <Tibor.Pittich@phuture.sk> \(85\)](#)
- [Till Kamppeter <till@mandrakesoft.com> \(289\)](#)
- [Tim Freeman <tim@infoscreen.com> \(4\)](#)
- [Tim Pickering <tim@as.arizona.edu> \(4\)](#)
- [Tim Pickering <tim@astro.rug.nl> \(8\)](#)
- [Tim Ricketts <tr@oxlug.org> \(15\)](#)
- [Tim Riker <Tim@Rikers.org> \(13\)](#)
- [Tim Westbrook \(2\)](#)
- [Timo Karjalainen <timok@iki.fi> \(6\)](#)
- [Toamsz Kłoczko <kloczek@rudy.mif.pg.gda.pl> \(63\)](#)
- [Tod Matola <matola@cvs.blackdown.org> \(8\)](#)
- [Todd Warner <taw@pobox.com> or <icode@pobox.com> \(1\)](#)
- [Tognon Stefano <ice00@users.sourceforge.net> \(12\)](#)
- [Tom Eastep <teastep@evergo.net> \(43\)](#)
- [Tom Faska <tom@ubertas.com> \(2\)](#)
- [Tom Rini <trini@kernel.crashing.org> \(6\)](#)
- [Tom Weber <x@4t2.com> \(3\)](#)
- [Tomasz Kłoczko <kloczek@idk.com.pl> \(7\)](#)
- [Tomasz Kłoczko <kloczek@rudy.mif.pg.gda.pl> \(1\)](#)
- [Tomsz Kłoczko <kloczek@idk.com.pl> \(3\)](#)
- [Tony Breiler \(barillo@telia.com\) \(2\)](#)
- [Tony Freitas \(tony@seacow.net\) \(2\)](#)
- [Tony Silva <tony_silva@alum.mit.edu> \(2\)](#)
- [Torsten Evers <tevers@ebh-interdata.de> \(4\)](#)
- [Torsten Naumann <Torsten.Naumann@Informatik.TU-Chemnitz.de> \(2\)](#)
- [Torsten Poulin <torsten@diku.dk> \(2\)](#)
- [Toshio Kuratomi <badger@prtr-13.ucsc.edu> \(15\)](#)
- [Travis Oliphant <Oliphant.Travis@altavista.net> \(3\)](#)
- [Travis Oliphant <Oliphant.Travis@mayo.edu> \(2\)](#)
- [Trevor Astrope <astrope@tabbweb.com> \(2\)](#)
- [Trevor Semeniuk <trevor@semeniuk.net> \(1\)](#)
- [Trinity College, Hartford, Connecticut, U.S.A. \(7\)](#)
- [Tristan Tarrant <nadir@users.sourceforge.net> \(16\)](#)
- [Troels Arvin <troels@arvin.dk> \(5\)](#)
- [Troll Tech <info@troll.no> \(2\)](#)
- [Trond Eivind Glomsrød \(teg@pvv.ntnu.no\) \(5\)](#)
- [Trond Eivind Glomsrød <teg@pvv.ntnu.no> \(42\)](#)
- [Troy Engel <tengel@sonic.net> \(134\)](#)
- [Tudor Hulubei <tudor@cs.unh.edu> \(3\)](#)
- [Turadg Aleahmad <turadg@guru.nu> \(2\)](#)
- [Turadg Aleahmad <turadg@guru.nu> \(2\)](#)
- [Ullrich Hafner <hafner@bigfoot.de> \(1\)](#)
- [Ullrich von Bassewitz <uz@musoftware.de> \(2\)](#)
- [Uniblax Consulting <jwhite@ghq.com> \(2\)](#)
- [Unknown \(36409\)](#)

- [Vaclav Ovsik <Vaclav.Ovsik@i.cz> \(16\)](#)
- [Vaclav Slavik <v.slavik@volny.cz> \(8\)](#)
- [Vaclav Slavik <vaclav.slavik@matfyz.cz> \(8\)](#)
- [Vadim Kurland <vadim@vk.crocodile.org> \(166\)](#)
- [Vadim Zaliva <lord@crocodile.org> \(125\)](#)
- [Vadim Zeitlin <vadim@wxwindows.org> \(60\)](#)
- [Vadim Zeitlin <zeitlin@dptmaths.ens-cachan.fr> \(9\)](#)
- [Valeriy Onuchin <onuchin@sirius.ihep.su> \(5\)](#)
- [Vanilla Server Development Team \(18\)](#)
- [Vasile Gaburici <gaburici@kermite.cs.pub.ro> \(5\)](#)
- [Venu Anuganti <venu@mysql.com> \(6\)](#)
- [Viggo.L.Norum@imm.ntnu.no \(2\)](#)
- [Ville SkyttÃ¤ <jpackage-discuss at zarb.org> \(17\)](#)
- [Ville SkyttÃ¤ <ville.skytta at iki.fi> \(265\)](#)
- [Ville SkyttÃ¤ <jpackage-discuss at zarb.org> \(3\)](#)
- [Ville SkyttÃ¤ <ville.skytta at iki.fi> \(76\)](#)
- [Vincent Danen <vdanen@mandrakesoft.com> \(129\)](#)
- [Vincent Guardiola <vguardiola@mandrakesoft.com> \(5\)](#)
- [Vincent Tassy <vince@butler-online.com> \(2\)](#)
- [Vitor Santos Costa <vitor@cos.ufrj.br> \(2\)](#)
- [Vlad Karpinsky <vlad@crocodile.org> \(10\)](#)
- [Vladimir Ivanov <vlad@elis.tasur.edu.ru> \(2\)](#)
- [Vladimir Ivaschekno <vi@maks.net> \(2\)](#)
- [Vladimir Ivaschenko <vi@maks.net> \(5\)](#)
- [Vladimir Prus <ghost@cs.msu.su> \(1\)](#)
- [Vladimir Támara <vtamara@users.sourceforge.net> \(1\)](#)
- [Vladyslav Zubkis cw2000@freenet.de \(2\)](#)
- [Vlatko Primorac <vlp@geocities.com> \(2\)](#)
- [Volker Kuhlmann <v.kuhlmann@elec.canterbury.ac.nz> \(13\)](#)
- [Volker Moell <moell@gmx.de> \(19\)](#)
- [Vu Hung Quan <binaire@videotron.ca> \(13\)](#)
- [W. L. Estes <wlestes@hamlet.uncg.edu> \(21\)](#)
- [W. Reilly Cooley <wcooley@nakedape.ml.org> \(1\)](#)
- [Wade Hampton <whampton@staffnet.com> \(2\)](#)
- [Walter Rawdanik <wrawdanik@netquotient.com> \(2\)](#)
- [Warly <warly@mandrakesoft.com> \(388\)](#)
- [Wayne Chang <wxc@iname.com> \(3\)](#)
- [Wayne Cuddy <wayne@crb-web.com> \(3\)](#)
- [Wayne Davison <wayned@users.sourceforge.net> \(27\)](#)
- [Wayne Dyer <dwdyer@eskimo.com> \(3\)](#)
- [Wemo Packager Manager <richwest@home.com> \(4\)](#)
- [Werner Bosse <wbosse@berlin.snafu.de> \(32\)](#)
- [Werner Bosse <wbosse@snafu.de> \(4\)](#)
- [Werner Bosse, <wbosse@berlin.snafu.de> \(2\)](#)
- [Willem Jan Palenstijn <wjpalenstijn@users.sourceforge.net> \(12\)](#)
- [William Evans <william.evans@computer.org> \(3\)](#)
- [William Stearns <wstearns@pobox.com> \(65\)](#)
- [William Stearns <wstearns@pobox.com>, Brian J. Murrell <brian@mountainviewdata.com>, and Gordon Matzigkeit <gord@fig.org> \(2\)](#)
- [William Stearns <wstearns@pobox.com>, and <kad@blackcatlinux.com> \(2\)](#)
- [Wilmer van der Gaast <lintux@dds.nl> \(2\)](#)

- [Wojciech Giziewicz \(wpgiziew@mit.edu\)](#) (4)
- [Wojciech Rawdanik <walter@netbyte2000.com>](#) (4)
- [Ximenes Zalteca <ximenes@mythic.net>](#) (3)
- [Ximenes Zalteca <ximenes@null.net>](#) (2)
- [XoXus <xoxus@usa.net>](#) (2)
- [Xose Vazquez <xose@wanadoo.es>](#) (3)
- [Yeechang Lee <ylee@columbia.edu>](#) (8)
- [Yellow Dog Linux <bugs@terraplex.com>](#) (1437)
- [Yoann Vandoorselaere <yoann@mandrakesoft.com>](#) (4)
- [Yoichi Imai <yoichi@silver-forest.com>](#) (13)
- [Yu-Chung Wang <wangy@ece.uci.edu>](#) (8)
- [Yuri Detzel <yad@mail.ru>](#) (2)
- [Yury Lebedev <yurylebedev@mail.ru>](#) (4)
- [Yves Duret <yduret@mandrakesoft.com>](#) (17)
- [Yves Duret <yves@zarb.org>](#) (45)
- [Zac Sprackett <zsprackett@valinux.com>](#) (4)
- [Zach Brown <zab@zabbo.net>](#) (1)
- [Zack Hobson <zack@malfunktion.net>](#) (3)
- [Zoltan Kota <zkota@gwdg.de>](#) (3)
- [Zoltan Varady <drifta@users.sourceforge.net>](#) (4)
- [Zoltán Kóta <zkota@gwdg.de>](#) (4)
- [Zoltán Kóta <kota@szbk.u-szeged.hu>](#) (1)
- [adilger@turbolinux.com](#) (6)
- [ajenkins@cs.umass.edu](#) (3)
- [aldug <aldug@openup.com>](#) (16)
- [andi payn <payn23@hotmail.com>](#) (1)
- [andi payn <payn@myrealbox.com>](#) (2)
- [andreas@gwydiondylan.org](#) (4)
- [andy@sower.manna.org](#) (1)
- [apt4rpm-devel@lists.sourceforge.net](#) (2)
- [bladder <bladder@ethernet.org>](#) (2)
- [bladder <bladder@pagan.org>](#) (10)
- [bri@biosci.umn.edu](#) (6)
- [bugs@redhat.com](#) (4)
- [cananian@alumni.princeton.edu,nathan.hand@anu.edu.au](#) (2)
- [cefiar <cefiar1@optushome.com.au>](#) (10)
- [checkinstall-1.3.0beta3](#) (2)
- [checkinstall-1.4.0beta2](#) (1)
- [checkinstall-1.4.0beta3](#) (2)
- [checkinstall-1.4.1](#) (1)
- [checkinstall-1.5.2](#) (37)
- [chorn@warwick.net](#) (6)
- [chrisa@asty.org](#) (67)
- [civileme <civileme@mandrakesoft.com>](#) (4)
- [cjean@nexen.net](#) (2)
- [claudio@conectiva.com](#) (3)
- [claudio@hellabs.org](#) (9)
- [cturner@redhat.com](#) (98)
- [cyclic@cyclicsyn.dimensiona.com](#) (2)
- [dam's <damien@mandrakesoft.com>](#) (4)
- [dams <dams@tuxfamily.org>](#) (4)

- daniel@veillard.com (2)
- [dann frazier <dannf@dannf.org>](mailto:dannfrazier@dannf.org) (121)
- dannybackx@users.sourceforge.net (1)
- dbarth@besancon.net (4)
- [diakka <diakka@resnet.gaech.edu>](mailto:diakka@resnet.gaech.edu) (3)
- [diakka <diakka@resnet.gatech.edu>](mailto:diakka@resnet.gatech.edu) (11)
- eric.kidd@pobox.com (3)
- ernasm@trabas.com (5)
- etienne@mandrakesoft.com (2)
- fastresolve-bugs@web.us.uu.net (2)
- feedback@suse.de (1698)
- [firedfly <firedfly@thestuff.net>](mailto:firedfly@thestuff.net) (2)
- fpays@club-internet.fr (2)
- gentsch@ifm.uni-hamburg.de (3)
- gerhard@bigfoot.de (8)
- [groomed <a.haakmat@chello.nl>](mailto:groomed@a.haakmat@chello.nl) (6)
- gt@esi.ac.at (8)
- h9304891@student.anu.edu.au (1)
- herbert@fh-wedel.de (1)
- <http://bugzilla.conectiva.com.br/> (1)
- <http://www.suse.de/feedback> (731)
- [iNOUE Koich! <inoue@ma.ns.musashi-tech.ac.jp>](mailto:iNOUE_Koich!@ma.ns.musashi-tech.ac.jp) (2)
- [iNOUE Koichi <inoue@ma.ns.musashi-tech.ac.jp>](mailto:iNOUE_Koichi@ma.ns.musashi-tech.ac.jp) (10)
- [ian geiser <geiseri@msoe.edu>](mailto:ian.geiser@msoe.edu) (10)
- [installwatch2rpm](#) (1)
- [jeff covey <jeff.covey@pobox.com>](mailto:jeff.covey@pobox.com) (6)
- jeff.covey@pobox.com (3)
- jhpb@sarto.gaithersburg.md.us (2)
- jlewis@cse.ogi.edu (2)
- [josh buhl <jbuhl@users.sourceforge.net>](mailto:josh.buhl@users.sourceforge.net) (10)
- jp@spektr.eu.org (4)
- kw@kwlux1 (2)
- lai@physics.utexas.edu (2)
- lindberg@id.wustl.edu & david@summersoft.fay.ar.us (1)
- lindberg@id.wustl.edu & mw@moni.msci.memphis.edu (1)
- [ljp <ljp@llornkcor.com>](mailto:ljp@llornkcor.com) (6)
- lord@jen.americas.sgi.com (34)
- manojk@io.com (1)
- marcsoft@magic.metawire.com (2)
- misha@nasledov.com (9)
- mjlong@mindspring.com (Michael J. Long) (1)
- [monkeyiq <monkeyiq@users.sourceforge.net>](mailto:monkeyiq@users.sourceforge.net) (2)
- moz@compsoc.man.ac.uk (6)
- mz@webideal.de (6)
- nathan.hand@anu.edu.au (13)
- nstraz@kali.americas.sgi.com (50)
- oracle@suse.com (1)
- pgodman@real.com (1)
- [pjohnsen <pelle.johnsen@mail.dk>](mailto:pjohnsen@pelle.johnsen@mail.dk) (1)
- platin@ms.ccafps.khc.edu.tw (2)
- ps@cruxware.org (3)

- python-ldap-dev@lists.sourceforge.net (1)
- rflynn@blueridge.net (12)
- rg@mbit-gmbh.de (1)
- riq@core-sdi.com (1)
- robin@gareus.de (1)
- root@chuckle.americas.sgi.com (14)
- root@kali.americas.sgi.com (6)
- root@linux-xfs2.americas.sgi.com (34)
- [rpm manager <rpm@eIT.de>](mailto:rpm_manager@eIT.de) (1)
- rpm@ied.com (3)
- rpotter@rpcs.net (1)
- [rufus t firefly <rufus.t.firefly@linux-mandrake.com>](mailto:rufus.t.firefly@linux-mandrake.com) (86)
- runnhide@runnhide.com (2)
- [slurp \(v 0.1.0\)](#) (2)
- [stimuli <stimulusresponse@hotmail.com>](mailto:stimuli@stimulusresponse.hotmil.com) (2)
- tanner@real-time.com (273)
- [technoir <technoir@themes.org>](mailto:technoir@themes.org) (1)
- [ufus t firefly <rufus.t.firefly@linux-mandrake.com>](mailto:rufus.t.firefly@linux-mandrake.com) (1)
- uz@musoftware.com (2)
- [vanRijn \(vR@movingparts.net\)](mailto:vR@movingparts.net) (6)
- [vsl](#) (2)
- wtanaka@users.sourceforge.net (24)
- [yoneda kenji <yoneda@n.isl.titech.ac.jp>](mailto:yoneda@n.isl.titech.ac.jp) (2)
- [Łukasz Trąbniński <lukasz@inx.pm.waw.pl>](mailto:lukasz@inx.pm.waw.pl) (2)
- [Łukasz Trąbniński <lukasz@lt.wsisiz.edu.pl>](mailto:lukasz@lt.wsisiz.edu.pl) (2)
- [Ørn E. Hansen](#) (8)

Generated by [rpm2html 1.8.1](#)

[Daniel Veillard](#), Thu Apr 15 16:05:03 2004